applied sciences

Special Issue Reprint

Design and Optimization of Manufacturing Systems

Edited by
Zoran Jurković, David Ištoković and Janez Gotlih

mdpi.com/journal/applsci

MDPI

Design and Optimization of Manufacturing Systems

Design and Optimization of Manufacturing Systems

Guest Editors

Zoran Jurković
David Ištoković
Janez Gotlih

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors
Zoran Jurković
Faculty of Engineering
University of Rijeka
Rijeka
Croatia

David Ištoković
Faculty of Engineering
University of Rijeka
Rijeka
Croatia

Janez Gotlih
Faculty of Mechanical Engineering
University of Maribor
Maribor
Slovenia

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Applied Sciences* (ISSN 2076-3417), freely accessible at: https://www.mdpi.com/journal/applsci/special_issues/KVC4J2E3E7.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-4407-4 (Hbk)
ISBN 978-3-7258-4408-1 (PDF)
https://doi.org/10.3390/books978-3-7258-4408-1

Contents

About the Editors

Zoran Jurković

Zoran Jurković is a Full Professor at the Faculty of Engineering, University of Rijeka, Croatia, where he works in the Department of Industrial Engineering and Management. His scientific and professional activities focus on industrial engineering, production management, and the digital transformation of production systems. He is particularly interested in developing and applying modern optimization and simulation methods to improve complex production processes. Prof. Jurković is currently involved in several scientific and professional projects to improve production efficiency and sustainability and to strengthen cooperation between the academic and industrial sectors. He is the author and co-author of numerous scientific and professional papers published in international journals and proceedings. He actively contributes, as a Guest Editor, to international journals in the field of mechanical engineering. His contribution has been recognized through awards for scientific and professional work as well as mentoring students and young researchers.

David Ištoković

David Ištoković is an Associate Professor at the Faculty of Engineering, University of Rijeka, Croatia, where he works in the Department of Industrial Engineering and Management. He obtained his Ph.D. in Mechanical Engineering in 2021, with a research focus on production planning and scheduling. His scientific and professional activities focus on the digital transformation of production systems, the application of digital twins, Industry 4.0, and the modeling and optimization of production processes. Dr. Ištoković is currently involved in several scientific research and professional projects dealing with the simulation, visualization, and optimization of complex systems in production. His scientific interests include the application of advanced optimization algorithms, the digitalization of processes, and the introduction of modern engineering tools into production and operational practice. He is the author and co-author of several scientific and professional papers and an active collaborator in industrial projects with the aim of transferring knowledge and technology from academia to industry, in order to optimize production and increase the competitiveness of companies.

Janez Gotlih

Janez Gotlih is an Assistant Professor at the Faculty of Mechanical Engineering, University of Maribor, Slovenia. He obtained his Ph.D. in Mechanical Engineering in 2020, with a research focus on robotic machining. He is actively engaged in student, industrial, and research projects at both national and international levels. Dr. Gotlih collaborates closely with academic and industrial partners and contributes to the wider scientific community through his participation in professional societies and as a Guest Editor for international journals in the field of mechanical engineering. His research interests lie in the development of intelligent manufacturing systems, with a particular focus on robotics, the optimization of manufacturing processes, advanced automation, and the integration of artificial intelligence in production environments.

Preface

This Special Issue reprint brings together a curated collection of contributions dedicated to the design, optimization, and advancement of manufacturing systems, reflecting current academic research and industrial development. The scope of this reprint includes topics such as intelligent scheduling and control, simulation-based optimization, integration of digital technologies, collaborative robotics, and human-centered and sustainable production. Our aim was to foster a multidisciplinary perspective that links theoretical foundations with practical implementations, thereby promoting innovative solutions to complex production challenges.

The motivation behind this Special Issue stemmed from the ongoing transformation of manufacturing systems under the influence of Industry 4.0 and digitalization. As Guest Editors, we sought to create a platform for researchers and practitioners to present high-quality work that addresses both strategic and operational aspects of modern manufacturing.

This reprint is intended for scholars, engineers, and decision-makers engaged in the research, design, and optimization of production systems. We would like to sincerely thank all the contributing authors for their excellent submissions and the reviewers for their constructive feedback, which ensured the high standard of the included works. We also extend our gratitude to the editorial team of *Applied Sciences* for their valuable support throughout the process.

Zoran Jurković, David Ištoković, and Janez Gotlih

Guest Editors

applied sciences

MDPI

Article

Application of Modified Steady-State Genetic Algorithm for Batch Sizing and Scheduling Problem with Limited Buffers

Gordan Janeš, David Ištoković *, Zoran Jurković * and Mladen Perinić

Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia
* Correspondence: distokovic@riteh.hr (D.I.); zjurkovic@riteh.hr (Z.J.)

Abstract: Batch sizing and scheduling problems are usually tough to solve because they seek solutions in a vast combinatorial space of possible solutions. This research aimed to test and further develop a scheduling method based on a modified steady-state genetic algorithm and test its performance, in both the speed (low computational time) and quality of the final results as low makespan values. This paper explores the problem of determining the order and size of the product batches in a hybrid flow shop with a limited buffer according to the problem that is faced in real-life. Another goal of this research was to develop a new reliable software/computer program tool in c# that can also be used in production, and as result, obtain a flexible software solution for further research. In all of the optimizations, the initial population of the genetic algorithm was randomly generated. The quality of the obtained results, and the short computation time, together with the flexibility of the genetic paradigm prove the effectiveness of the proposed algorithm and method to solve this problem.

Keywords: hybrid flow shop; batch size; scheduling; buffer configuration; optimization; steady-state genetic algorithm

1. Introduction

A genetic algorithm (GA) is a heuristic search that mimics the process of natural evolution. The use of genetic algorithms for optimization was first introduced by Holland [1]. It is a stochastic heuristics technique which encompasses a semi-random search method whose mechanism is based on the evolutionary processes that occur in nature. The search method in a genetic algorithm is not based on improving a single solution, and instead, a genetic algorithm works with multiple possible solutions. The GA as a search mechanism is usually used when there is very little knowledge of the solution space or when there are far too many possible solutions to use the standard search/optimization methods.

Many commonly used genetic algorithms have a major drawback: they have high computational power requirements and some have high demands on the CPU or memory of it, or usually, both, so a modified genetic algorithm based on a steady-state genetic algorithm (SSGA) was developed with the following goals:

- To have reliable results;
- To be fast;
- To be suitable to run in a highly parallelized computer environment;
- To have low demands on the computer memory;
- Keep a low number of genetically identical individuals;

Based on the chosen algorithm, completely new software for testing was written in C#.

The schedule and size of the product batches and buffer configuration are among the major problems in manufacturing since manufacturing systems in a real environment have frequent requirements for change. This is mainly due to today's turbulent manufacturing environment calls for adaptive and rapidly responding manufacturing systems because there is a high level of market competition. Other reasons for this can be problems in the supply of raw material or energy and there being a long downtime due to the malfunction of production equipment. So, flexibility and re-configurability are becoming more and more important in modern manufacturing [2]. Those who are responsible for making strategic decisions in manufacturing companies must have relevant pieces of information as soon as possible to make relevant decisions on how to use the available resources and retain market competitiveness.

The buffer storage in production serves to decrease and balance the processing times, increase the production flexibility and decrease the impact of the breakdowns in production, but they are at the cost of additional capital investment, the floor space of the line, and using more inventory. Hence, the determining size and allocation of the buffer storage in manufacturing systems has both theoretical and practical interests.

In this paper, the performance of a specially developed software-based modified genetic algorithm was tested on the real-life problem of determining the size and schedule of the product batches with multiple buffer storage configurations in a hybrid flow shop (HFS). The study showed that the proposed approach was highly effective in the finding acceptable solutions for all of the problem sets that were examined.

2. Literature Review

A large number of studies can be found in the literature which solve the problem of determining the batch size and schedule in a HFS. According to Ribas et al. [3], a HFS actually represents the production systems where a larger range of products move unidirectional through several production stages. In each production stage, there are one or more identical production capacities. A literature review on hybrid flow shop scheduling problem was given by Ruiz and Vázquez-Rodríguez [4] and Tosun et al. [5]. This type of manufacturing system is characteristic of many processes and discrete manufacturing companies from the real environment [6], such as automobile manufacturing, machinery manufacturing, etc. For this reason, this type of manufacturing system was chosen as the subject of research.

A HFS scheduling problem is a complex combinatorial problem. There are many different variations of this problem, i.e., it differs depending on several factors, such as the understanding of the environmental assumptions, constraints or performance measures that are to be achieved [7]. For example, many authors have used different methods to minimize the makespan [8–11]. Makespan is one of the most frequently used performance measures because makespan directly affects the performance of the manufacturing efficiency in the form of there being satisfying delivery that is on time and reducing the production costs. Additionally, other performance measures have been observed, as can be seen detailed in [12,13].

Market uncertainty and a change in the production philosophies, where the stocks represent direct losses for many processes and discrete manufacturing companies, necessitate the production of a wide range of products within the timeframe. This leads to an increasing challenge in terms of determining the batch size and schedule. With this realization, the research attention focused on making both decisions at the same time has increased [14]. In addition, there is an increasing number of papers that deal with these problems using real-world examples [15–17].

Besides batch sizing and scheduling, the problem of accumulating product units between the manufacturing stages is also a major problem. Therefore, it is important to know what capacity of buffer storage to provide for the smooth running of the production. Leisten [18] spoke about the influence of a limited buffer in 1990. Makespan minimization problems for a two-stage flow shop with a limited buffer were considered in [19,20]. Jiang and Zhang [21] were investigated an energy-oriented scheduling problem deriving from a hybrid flow shop with limited buffers. A solution for multi-objective permutation flowshop scheduling problems with limited buffers was presented by Qian et al. They proposed an effective hybrid algorithm based on differential evolution [22]. For the same type of problem, Liang et al. [23] developed a multi-objective hybrid self-adaptive differential evolution optimization algorithm. Zohali et al. [24] developed a metaheuristic algorithm, which was called the discrete fruit fly algorithm, to solve a batch scheduling problem in a hybrid flow shop with limited buffers. Marinelli et al. [25] dealt with capacitated batch sizing and a scheduling problem with parallel machines and shared buffers. Batch sizing and scheduling problems with buffer constraints were also presented by Sundaramoorthy and Maravelias [26].

For solving the batch sizing and scheduling problems and buffer configuration problems, one of the most commonly used methods and techniques is GA. GA has proven to be a relatively good optimization tool. An example of that is given by Shen, K. et al. [16]. Han et al. [17] proposed an improved compact genetic algorithm in a hybrid flow shop with a multi-queue buffer. Amjad et al. developed and implemented a modified GA for makespan optimization, where the GA was initialized based upon global, local and random selection techniques, and adaptive reproductive operators were applied to intelligently evolve the algorithm [27]. For solving the multi-objective optimization problem in HFS, Chen and Zhao [28] introduced new constraints and improved the traditional GA. By combining a Random Key Genetic Algorithm (RKGA) and a Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS), Karacan et al. [29] proposed a new integrated methodology.

However, many of the proposed ones do not give results enough fast or some modifications are needed to ensure that it is easier to apply them. Therefore, this paper proposes a modified steady-state GA to solve this type of problem, which will serve as a kind of tool for the fast and efficient determination of the batch size and schedule.

So, the first step in this study was to create a modified GA that would be easy to apply and give results quickly. Additionally, the second step was to apply the modified GA to a real-world example with the aim to determine the batch size and schedule and the required buffer configuration.

3. The problem Formulation

3.1. Notation

The production system and all the data used in this article are based on data from the real world. The problems of the manufacturing company are to find the appropriate batch size for each product type, to order the allocation of the production capacities to each batch of products and to determine the necessary buffer configuration to ensure that smooth production occurs and to ensure that the delivery is on time.

The parameters used in this paper are given in Appendix A.

3.2. Description

The plant is producing groups of technologically similar products. The technologically similar products are those that have a high degree of similarity in the order of processing and duration of the operations. In this case, the plant is producing three types of technologically similar products (which are labelled as D, E and F) and the delivery is scheduled every two weeks. The targeted two-week production for each product is given in Table 1 as q_j (q_D, q_E, q_F).

Table 1. Two-week production goal.

j	D	E	F
q_j	4616	3232	2616

The working week lasts for five days and in two shifts. Each shift lasts eight hours, so the maximum availability of the production equipment, in this paper, was 160 h (C_{goal} = 160 h). The makespan C_{calc} must be lower than the maximum availability of the production equipment (1) or there will be a delay in the delivery, which should definitely be avoided.

$$C_{\text{calc}} < C_{\text{goal}}, \quad (1)$$

Delivering on time is a condition that has to be satisfied, but it is not required that the makespan must be as small as possible. Every value that satisfies that the delivery is on time is acceptable. The processing times p_{ijk} for the operations o_{ij} are given in Table 2.

Table 2. Processing times (in hours).

o_{ij}	D	E	F	M_i
1	0.028	0.035	0.036	3
2	0.031	0.035	0.036	3
3	0.012	0.010	0.010	1
4	0.020	0.019	0.017	2
5	0.008	0.012	0.014	1

This production system is typically a hybrid flow shop. Precisely, this production system has five stages, where each stage consists of the M_i number of the same machines. All three types of products pass unidirectionally through the production system and are processed at each stage.

The arrival time of each batch at the stage where the type of product that is produced is different from the previously processed one, and this means that there is a need to setup the workplace. The setup times are defined as being relatively long, in this case, this is mainly because of the client's request. Since, there are technologically similar products, the same setup time was used for all of the possible combinations of product changes at the same stage. Therefore, the setup time ST_i for each stage is defined, as can be seen in Table 3.

Table 3. Setup times (in hours).

i	1	2	3	4	5
ST_i	0.233	0.233	0.267	0.267	0.267

As mentioned before, the products travel in batches through the production system. Due to the simplicity of production management and control, the quantity of products in each batch of the same type of product B_{jb} is set as equal (2). Except in the last batch B_{jl}, where the quantity of product is equal to the remaining difference between the production goal of the same type of product and the quantity of product that has been produced so far (3). The order in which the batches of products enter the system is not defined, but it is completely random.

$$B_{j1} = B_{j2} = \ldots = B_{j(l-1)}, \quad (2)$$

$$B_{jl} = q_j - B_j \cdot (l - 1), \quad (3)$$

After finishing the processing at the previous stage, the batch is sent directly to the next stage if that stage is not occupied. Otherwise, the previous stage is blocked. For this reason, a buffer f is needed between each stage that can receive a certain amount of product. There are a total of four buffers where each can receive a different amount of product. It is worth noting that several different batches can be on the same buffer at the same time. The buffer can be recharged until the moment that it can no longer receive the entire batch. In that case, the batch is waiting in the previous stage for the space in the buffer to be freed up. After the next stage is empty (releasing the previous batch), then it is occupied by the batch that first arrived on that buffer (according to the FIFO principle).

The batch size directly affects the required buffer size. On the other hand, the buffer size takes up space in the production plant. When the problem of batch scheduling is added to all this, it is necessary to adjust these values in order to achieve optimal results. First of all, this is performed to satisfy the on-time delivery demand.

In the first step, it is necessary to determine whether the defined production goals can be produced. If it is possible to deliver them on time, we must determine which buffer configuration is required at which position. After that, it is necessary to determine the batch size and the order for the proposed buffer configurations, and observe which one gives the best result in terms of satisfying that the delivery is on time.

In the optimization process, the following statements were also taken into account:

- All stages, machines and buffers are available at time 0;
- All jobs are released at time 0;
- Every operation is a part of a chain of operations;
- The order of operations must be followed;
- All of the operations of a given job have to be processed in a given order;
- Two operations of a job cannot be processed at the same time;
- Each machine can process at most one operation at a time;
- Once processing starts on a given machine, it must complete on that particular machine without any interruption;
- The utilization of each machine is set as 0.85;
- Each operation has a fixed duration.

4. Genetic Algorithms Parameters

4.1. Initial Population

In all of the calculations for this paper, the initial population has been created completely randomly. Such an approach in the process of creation of the initial population, in some cases, can result in a longer search time and require more computational resources, but this approach provides the initial population with a high genetic diversity and a larger pool of genetic material. Such as in real life, in nature, a high level of the genetic diversity of a certain species increases the likeliness that that species will find (better) solutions for the challenges of evolution. Therefore, population sizes of 100 individuals were used.

4.2. Coding and Decoding of Organisms (Solutions)

The first step in constructing a genetic algorithm is to define an appropriate genetic representation and coding. Choosing a good representation is very important because this step significantly affects all of the other steps in the algorithm, and consequently, it has a big impact on the quality of results and speed of the algorithm. In determining the genetic representation, the goal was also to obtain flexible code representation for further research and software development.

A set of three 2D (two-dimensional) integer arrays were used for the chromosome representation:

- Product sequence array—Integer type;
- Batch quantity array—Integer type;
- Fitness array—(real number) double-precision floating-point type.

The usage of one 3D (three-dimensional) array is less challenging from a programmer's point of view, but with the 2D arrays, the computer program runs significantly faster. Before the final decision could be made on how to define the genetic representation, another computer program was written to test memory operations speed, and the test showed that usage of 2D arrays runs approx. 20 percent faster.

4.3. Solution Decoding

For better software efficiency, the GA population (possible solutions) were coded, and at the end of the GA, the best solution has to be converted into a more readable format. The decoding was performed by connecting the corresponding values from two arrays—Figure 1. The first array contains the product designation and the second batch size. By combining these two values, we obtain results that are in an easily readable format.

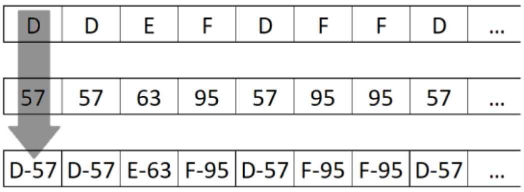


Figure 1. Decoding.

4.4. Crossover (and Selection)

In the genetic algorithm that was used in this research, the processes of selection and crossover are combined — Figure 2. This approach is inspired by nature: a habitat with strict boundaries can support only a certain number of species, and this number is limited by the habitat's resources, and also, the species tend to grow until the habitat's limit is reached. Individuals that are more adjusted to the environment tend to outlive the individuals that are less adjusted to the habitat parameters. As it was inspired by this idea, in this algorithm, there is no selection for a standalone GA operator in which part of the population is removed, and instead, the removal of the individuals is integrated with the crossover GA operator, and only one individual is removed in each step. The emptied space is fulfilled with an individual that is created during the crossover operation. Also, during this process, the removal of identical individuals is integrated to keep the genetic diversity within the population as high as possible. If two parents that are selected for the crossover operation have the same fitness, then their genome is compared, and if they are identical, one of them is removed from the population to make space for the new individual that will be created in the crossover process.

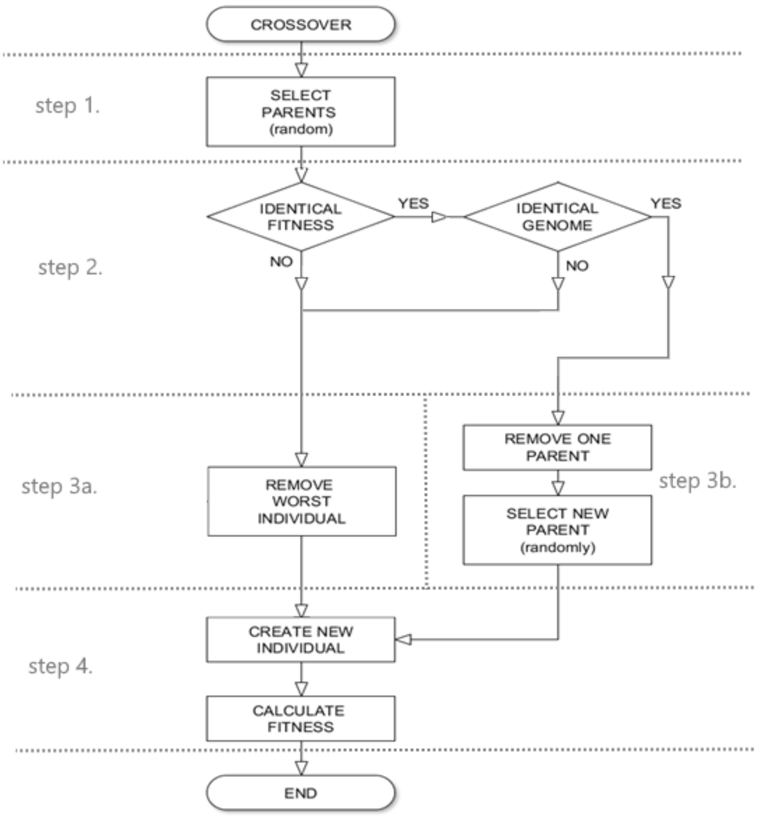


Figure 2. Combined processes of crossover and selection.

Step 1: In the first step, two individuals from the population are selected, and they will be used as the parents. Their genetic material will be used in the process of the crossover and the creation of a new individual. The selection of parents is completely random, and all of the individuals in the population have equal chances to be selected.

Step 2: In this step, the fitness of two individuals that were selected as parents are compared. In the first stage of comparison, their fitness is compared, and if they do not have the same fitness, they do not have an identical genome, and their genetic material can be used in the creation of a new individual, and so the algorithm continues on to step 3a. However, if they have the same fitness, an additional check is made, and their complete genome is compared. In the case of two non-identical individuals, the algorithm continues on to step 3a. Otherwise, the algorithm continues on to step 3b.

Step 3a: Since the parents are not identical, the individual with the worst fitness is removed from the population to make space for the individual that will be created in step 4. (Since it has the lowest fitness in the population, it is most likely that the usage of genetic material from this individual will not produce competitive offspring.)

Step 3b: Because in step 2, two genetically identical parents were selected. The result of the crossover process would be the creation of a new individual that is genetically identical to its parents. To avoid this, one of the parents is removed from the population, and randomly selected individuals take its place. This mechanism not only prevents the creation of identical individuals, but it can also remove individuals with identical genomes. The usage of this mechanism also prevents the loss of genetic material (and consequently, it increases the chance to obtain better final algorithm results).

Step 4: The new individual is created as a result of the combination of the genetic material of its parents, and the newly created individual takes the place of the removed one, and the population again has a size determined by the habitat. After the new individual is created, its fitness is calculated, and if it has the best or worst fitness in the population, it is marked as such. This is the approach to selection in the genetic algorithm:

- It ensures the minimum loss of material from the genetic pool, and consequently, it has a higher chance to produce better results;
- It has minimum requirements towards the computer memory since needs space for only one population in computer memory;
- There is no need to sort the population-based on fitness and/or to make many comparisons of fitness within the population.

4.5. Mutation

A mutation introduces new genetic material into the population to enhance the diversity of a population. In GA, a mutation is applied with a small probability since a large probability of a mutation may lead to loss of good genetic material, and as a result, a downgrade in the quality of results, and it may make the algorithm slower.

In this case, there are two possible types of mutation: the change of the batch order or the change of the batch size. In the first case, the case change of the batch order mutation process is fairly simple. Only two genes replace their position. However, in the case of a change of the batch size, additional modifications are required. To maintain the predefined quantity of the product, in most cases, it is necessary to correct the number of batches and size of the last batch. Batches that had to be added or removed and their position are determined randomly.

This approach to the genetic algorithm enables us to make multiple mutations (modification of genome) and/or multiple selections and crossover operations at the same time.

4.6. Definition of the Fitness Function

After the creation of the initial population, the fitness of each individual in the population C_{fitness} (4) is calculated.

$$C_{\text{fitness}} = C_{\text{calc}} + C_{\text{penaltyl}}, \tag{4}$$

Since, in this case, the delivery condition has to be satisfied, penalization of the possible solutions (5) that do not satisfy this condition was introduced into the fitness formula as C_{penalty}.

$$
\begin{aligned}
& C_{\text{calc, penalty}} = C_{\text{calc}} - C_{\text{goal}} \\
& \text{if } C_{\text{calc, penalty}} > 0 \\
& \quad C_{\text{penalty}} = (1 + C_{\text{calc, penalty}}) \cdot (C_{\text{calc}} - C_{\text{goal}}) \\
& \text{else} \\
& \quad C_{\text{penalty}} = 0 \\
& \text{end,}
\end{aligned}
\tag{5}
$$

5. Results and Discussion

5.1. Computational Time

The computer program was launched 100 times with different population sizes on the personal computer with AMD Ryzen 7 2700X Eight-Core Processor at 3.70 GHz CPU, 16 GB RAM and Windows 10 operating system. The population size was 100 individuals. The mutation rate was 2%, and the crossover rate was 75 % of the population (the number of selection/crossover operations in each generation was 0.75 of the population number) in 200 generations. The average running time is shown in Table 4. The software has two modes: "normal" and "log". In the "log" mode, the software keeps logs of some operation/calculation, and that makes it approximately 30 % slower. The values in Table 4 were achieved in the "normal" mode.

Table 4. Running time.

		Population Size [pcs]				
		50	100	200	300	500
Generations	100	00:09.273	00:20.211	00:44.316	01:01.513	01:46.551
	200	00:12.772	00:35.135	01:26.321	02:17.244	03:00.225

Increasing the GA parameters over 200 generations and 100 individuals in the population did not produce better results, and the calculation time was still acceptable, so these GA parameters were used in the further calculation.

For the comparison, the results and speed of the steady-state algorithm (SSga) that was used in this research were compared with the steady-state generation algorithm (SSGga). After consecutive 100 runs, the average calculated makespan value was lower (better) with the steady-state (SSga) algorithm. On the other hand, the steady-state generation algorithm (SSGga) had a better performance regarding the software run-time, as shown in Table 5.

Table 5. Makespan and running time (SS-GA and SSG-GA).

Algorithm	avg. Time [sec]	avg. Makespan [h]
steady-state algorithm (SS-GA)	36.778	160.087
steady-state generation algorithm (SSG-GA)	19.181	162.103

Despite the much lower computational time of the steady-state generation algorithm, priority was given to the quality of the results of the steady-state algorithm because the duration of the execution is acceptable in both cases.

5.2. Results

The software was also used to test multiple real-world problems in the industry using multiple scenarios. The research was conducted in two stages:

- The analysis of the need for a buffer in production;
- The analysis of the possible buffer configuration scenarios.

In the first step, a set of simulations was conducted to find where the greatest need for a buffer between the operations was. So, in this set of simulations, a GA without buffer limitations was executed 100 times. The ten best results were taken into consideration to find out the buffer needs (Table 6). The buffer is shown as a number of pieces, and since all three products are similar, the buffer capacity is same for all three products.

Table 6. Required buffer configuration.

C_{fitness}	B_D	B_E	B_F	F_{1_MAX}	F_{2_MAX}	F_{3_MAX}	F_{4_MAX}
155.610	119	295	164	595	238	164	357
156.534	123	225	159	680	434	159	492
156.827	161	156	147	591	322	156	322
157.173	230	267	170	526	267	198	340
157.273	229	182	253	676	265	182	482
157.309	224	200	300	704	360	216	448
157.314	201	233	255	605	267	66	468
157.334	137	135	209	643	369	137	542
157.455	184	260	175	568	260	184	368
157.603	178	126	157	544	178	157	304
			Average:	613.2	296.0	161.9	412.3

As can be seen in the last row in Table 6, the requirements for the buffer are the largest between the first and second operations and between the fourth and fifth production stages. Otherwise, the smallest buffer requirements is between the third and fourth production stages. These results were taken into consideration for the next stage, i.e., determining the optimal buffer configuration.

In the second stage, we tested six possible buffer configurations (BC1-BC6). The configurations were determined according to the data from Table 6, and the situations in the production facility and six possible configurations were taken into consideration (Table 7). For each scenario, the GA was executed 100 times with the same parameters as in 5.1.

Table 7. Buffer configurations and results.

Buffer Configuration	F_1	F_2	F_3	F_4	B_D	B_E	B_F	Average C_{fitness}	Best C_{fitness}
BC1	90	90	90	90	87	86	75	163.787	159.038
BC2	90	60	90	120	60	60	59	171.562	166.604
BC3	90	90	60	120	54	60	60	171.353	165.592
BC4	120	60	90	90	56	60	57	171.526	165.842
BC5	120	90	60	90	60	60	59	171.562	166.604
BC6	120	60	60	120	60	60	53	170.464	165.383

According to the simulation results, the best makespan was achieved with the BC1 buffer configuration, both in an average value for 100 consecutive simulations and with the best overall result. The BC1 buffer configuration is also the only buffer configuration that meets the specified condition of there being an on-time delivery.

6. Conclusions and Future Research

In this paper, a modified steady-state genetic algorithm was tested to solve the optimization problem. The problem of determining the batch order and the size of three technologically similar products in a hybrid flow shop with a limited buffer capacity was taken from the real environment. The results that were obtained in chapter 5 of this research show that both the performance and speed of this algorithm are quite good, and that the development of highly specialized and custom-made software can be taken into consideration during the process of production planning.

In short, the low execution time of the algorithm allows for the testing of many possible scenarios, and consequently, it obtains a better production process configuration. The results also show that the software based on the developed algorithm can serve as a reliable tool for its use in production planning. This can be very important in cases where there is not much available time to make new production configurations from scratch or to reconfigure existing ones.

Although genetic algorithms have been recognized as effective search algorithms for many years, continuous improvements in computer performances open new possibilities for the further development of genetic algorithms. In future research, the performance of the genetic algorithm that is used in this paper will be tested on even more complex problems to prove the capability of the algorithm to obtain reliable results.

The GA that was developed and used in this research is very suitable for parallelization and its use on modern multicore CPUs. It can be assumed that the parallelization of the computer code for the proposed genetic algorithm could bring significant performance improvements, and that the algorithm can work on highly parallelized computing systems. In this paper, we used a single thread solution, and so, a further reduction of the computational time is possible with some code modifications being performed. In further research, it is possible to investigate single multi-tread vs. multiple single thread solutions.

Author Contributions: Conceptualization, G.J. and D.I.; methodology, D.I.; software, G.J.; validation, G.J., D.I. and Z.J.; formal analysis and data curation, M.P.; resources, Z.J. and M.P.; writing—original draft preparation, G.J. and D.I.; writing—review and editing, G.J., D.I. and Z.J.; investigation, D.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by University of Rijeka, Croatia (contract number uniri-tehnic-18-223 and uniri-tehnic-18-100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Details regarding the data can be obtained by emailing the corresponding author.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Parameters Used in This Paper

N	set of jobs/products
S	set of stages
M_{i}	set of the parallel machines at stage i
L	set of batches of the same product type
H	set of buffers
j	job/type of product, j = 1, . . . , n
i	stage, i = 1, . . . , s
k	machine, k = 1, . . . , m
b	batch of the same product type, b = 1, . . . , l
f	buffer, f = 1,.., h
o_{ij}	operation of product j at stage i
p_{ijk}	processing time of product j at stage i on machine k
ST_{ijk}	setup time of product j at stage i on machine k

B_j	batch size of product j
B_{jb}	batch size of batch b of product j
q	production goal/quantity
F	buffer configuration/capacity

References

1. Holland, J.H. Genetic algorithm. *Sci. Am.* **1992**, *267*, 66–73.
2. Järvenpää, E.; Luostarinen, P.; Lanz, M.; Tuokko, R. Adaptation of Manufacturing Systems in Dynamic Environment Based on Capability Description Method. In *Manufacturing System*; Aziz, F.A., Ed.; IntechOpen: London, UK, 2012; pp. 93–118. Available online: https://www.intechopen.com/chapters/36406 (accessed on 3 October 2022). [CrossRef]
3. Ribas, I.; Leisten, R.; Framiñan, J.M. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Comput. Oper. Res.* **2010**, *37*, 1439–1454. [CrossRef]
4. Ruiz, R.; Vázquez-Rodríguez, J.A. The hybrid flow shop scheduling problem. *Eur. J. Oper. Res.* **2010**, *205*, 1–18. [CrossRef]
5. Tosun, Ö.; Marichelvam, M.K.; Tosun, N.A. Literature review on hybrid flow shop scheduling. *Int. J. Adv. Oper. Manag.* **2020**, *12*, 156–194. [CrossRef]
6. Framinan, J.M.; Leisten, R.; Ruiz García, R. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*; Springer: London, UK, 2014. [CrossRef]
7. Lee, T.-S.; Loong, Y.-T. A review of scheduling problem and resolution methods in flexible flow shop. *Int. J. Ind. Eng. Comput.* **2019**, *10*, 67–88. [CrossRef]
8. Ernst, A.; Fung, J.; Singh, G.; Zinder, Y. Flexible flow shop with dedicated buffers. *Discret. Appl. Math.* **2018**, *261*, 148–163. [CrossRef]
9. Lo, T.-C.; Lin, B.M.T. Relocation Scheduling in a Two-Machine Flow Shop with Resource Recycling Operations. *Mathematics* **2021**, *9*, 1527. [CrossRef]
10. Ren, J.F.; Ye, C.M.; Li, Y. A new solution to distributed permutation flow shop scheduling problem based on NASH Q-Learning. *Adv. Prod. Eng. Manag.* **2021**, *16*, 269–284. [CrossRef]
11. Zheng, J.; Wang, Y. A Hybrid Bat Algorithm for Solving the Three-Stage Distributed Assembly Permutation Flowshop Scheduling Problem. *Appl. Sci.* **2021**, *11*, 10102. [CrossRef]
12. Shen, C.; Chen, Y.L. Blocking flow shop scheduling based on hybrid ant colony optimization. *Int. J. Simul. Model.* **2020**, *19*, 313–322. [CrossRef]
13. Cheng, C.-Y.; Lin, S.-W.; Pourhejazy, P.; Ying, K.-C.; Lin, Y.-Z. No-Idle Flowshop Scheduling for Energy-Efficient Production: An Improved Optimization Framework. *Mathematics* **2021**, *9*, 1335. [CrossRef]
14. Wörbelauer, M.; Meyr, H.; Almada-Lobo, B. Simultaneous lotsizing and scheduling considering secondary resources: A general model, literature review and classification. *OR Spectr.* **2019**, *41*, 1–43. [CrossRef]
15. Yurtkuran, A.; Yagmahan, B.; Emel, E. A novel artificial bee colony algorithm for the workforce scheduling and balancing problem in sub-assembly lines with limited buffers. *Appl. Soft Comput.* **2018**, *73*, 767–782. [CrossRef]
16. Shen, K.; De Pessemier, T.; Martens, L.; Joseph, W. A parallel genetic algorithm for multi-objective flexible flowshop scheduling in pasta manufacturing. *Comput. Ind. Eng.* **2021**, *161*, 107659. [CrossRef]
17. Han, Z.; Zhang, Q.; Shi, H.; Zhang, J. An Improved Compact Genetic Algorithm for Scheduling Problems in a Flexible Flow Shop with a Multi-Queue Buffer. *Processes* **2019**, *7*, 302. [CrossRef]
18. Leisten, R. Flowshop sequencing problems with limited buffer storage. *Int. J. Prod. Res.* **1990**, *28*, 2085–2100. [CrossRef]
19. Zhang, C.; Shi, Z.; Huang, Z.; Wu, Y.; Shi, L. Flow shop scheduling with a batch processor and limited buffer. *Int. J. Prod. Res.* **2016**, *55*, 3217–3233. [CrossRef]
20. Zhang, G.; Xing, K. Differential evolution metaheuristics for distributed limited-buffer flowshop scheduling with makespan criterion. *Comput. Oper. Res.* **2019**, *108*, 33–43. [CrossRef]
21. Jiang, S.-L.; Zhang, L. Energy-Oriented Scheduling for Hybrid FlowShop With Limited Buffers Through Efficient Multi-Objective Optimization. *IEEE Access* **2019**, *7*, 34477–34487. [CrossRef]
22. Qian, B.; Wang, L.; Huang, D.; Wang, W.; Wang, X. An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Comput. Oper. Res.* **2009**, *36*, 209–233. [CrossRef]
23. Liang, J.; Wang, P.; Guo, L.; Qu, B.; Yue, C.; Yu, K.; Wang, Y. Multi-objective flow shop scheduling with limited buffers using hybrid self-adaptive differential evolution. *Memetic Comput.* **2019**, *11*, 407–422. [CrossRef]
24. Zohali, H.; Naderi, B.; Mohammadi, M. The economic lot scheduling problem in limited-buffer flexible flow shops: Mathematical models and a discrete fruit fly algorithm. *Appl. Soft Comput.* **2019**, *80*, 904–919. [CrossRef]
25. Marinelli, F.; Nenni, M.E.; Sforza, A. Capacitated lot sizing and scheduling with parallel machines and shared buffers—A case study in a packaging company. *Ann. Oper. Res.* **2007**, *150*, 177–192. [CrossRef]
26. Sundaramoorthy, A.; Maravelias, C.T. Modeling of Storage in Batching and Scheduling of Multistage Processes. *Ind. Eng. Chem. Res.* **2008**, *47*, 6648–6660. [CrossRef]

27. Amjad, M.K.; Butt, S.I.; Anjum, N.; Chaudhry, I.A.; Faping, Z.; Khan, M. A layered genetic algorithm with iterative diversification for optimization of flexible job shop scheduling problems. *Adv. Prod. Eng. Manag.* **2020**, *15*, 377–389. [CrossRef]
28. Chen, D.; Zhao, X.R. Production management of hybrid flow shop based on genetic algorithm. *Int. J. Simul. Model.* **2021**, *20*, 571–582. [CrossRef]
29. Karacan, I.; Karacan, I.; Senvar, O.; Bulkan, S. An Integrated Solution Approach for Flow Shop Scheduling. *Teh. Gaz.* **2021**, *28*, 786–795. [CrossRef]

applied sciences

MDPI

Review

Task Complexity and the Skills Dilemma in the Programming and Control of Collaborative Robots for Manufacturing

Peter George, Chi-Tsun Cheng *, Toh Yen Pang and Katrina Neville

School of Engineering, STEM College, RMIT University, Melbourne, VIC 3000, Australia
* Correspondence: ben.cheng@rmit.edu.au

Abstract: While traditional industrial robots participate in repetitive manufacturing processes from behind caged safety enclosures, collaborative robots (cobots) offer a highly flexible and human-interactive solution to manufacturing automation. Rather than operating from within cages, safety features such as force and proximity sensors and programmed protection zones allow cobots to work safely, close to human workers. Cobots can be configured to either stop or slow their motion if they come in contact with a human or obstacle or enter a protection zone, which may be a high pedestrian traffic area. In this way, a task can be divided into sub-processes allocated to the cobot or the human based on suitability, capability or human preference. The flexible nature of the cobot makes it ideal for low-volume, 'just-in-time' manufacturing; however, this requires frequent reprogramming of the cobot to adapt to the dynamic processes. This paper reviews relevant cobot programming and control methods currently used in the manufacturing industry and alternative solutions proposed in the literature published from 2018 to 2023. The paper aims to (1) study the features and characteristics of existing cobot programming and control methods and those proposed in the literature, (2) compare the complexity of the task that the cobot is to perform with the skills needed to program it, (3) determine who is the ideal person to perform the programming role, and (4) assess whether the cobot programming and control methods are suited to that person's skillset or if another solution is needed. The study is presented as a guide for potential adopters of cobots for manufacturing and a reference for further research.

Keywords: cobot; collaborative robot; programming; control; skills; task complexity; teach pendant; manufacturing

1. Introduction

Cobots are an integral part of modern manufacturing, destined to become the leading form of robotics technology in the future [1]. While most mechanical equipment currently used in manufacturing is mono-functional, cobots can assist with a wide variety of tasks, including some standard operations listed in Table 1, along with many more custom applications. As systems incorporate self-learning capabilities of artificial intelligence and auto-correction [2] into cobot operations, the subsequent autonomous behaviour can provide unprecedented collaborative assistance to human workers [3,4]. Evolving from traditional industrial robots [5], cobots have additional programming requirements to feed a collaborative functionality, as shown in Table 2.

Table 1. Typical cobot tasks (Adapted from [6]).

Cobot Task	General Application Description
Welding	Joining of metal parts, typically with a MIG welding (automatic welding wire feed) process
Machining	Precision surfacing process involving a milling or cutting tool

Table 1. *Cont.*

Cobot Task	General Application Description
Deburring	Removal of waste material from casting or machining processes with an abrasive tool
Polishing	Treatment to remove surface irregularities or attain a lustre to coated or machined surfaces
Spray Painting	Part coating, applied to protect or otherwise enhance surface appearance
Sorting	Practical categorisation of unsorted parts for kitting, classification or other organisational operation
Pick & Place	Moving components from a starting point to an endpoint for assembly or other processes
Stacking	Moving finished products from a production line to a pallet or other storage location
Machine Tending	Inserting billets or parts into a milling machine, lathe, etc. and retrieving machined or processed components
Inspection/Measurement	Analysis or quality assurance process where parts and other sub-assemblies or components are measured with sensors or by other means to ensure they are within an acceptable tolerance range

Table 2. Comparison between industrial robot and cobot capabilities [7–9].

Feature	Traditional Industrial Robot	Collaborative Robot
Engagement with humans	Segregated. Operates within a protective barrier, away from humans	Interactive. Operates collaboratively with humans
Safety near humans	Not safe. Must work separately	Safe to work with humans
Environmental awareness	Cannot dynamically adapt behaviour	Can adapt to changes in the environment
Programming flexibility and complexity	Fixed, rudimentary use case programs. Typically reprogrammed infrequently	Flexible, customised use case programs. Potentially reprogrammed frequently
Ease of implementation	Arduous. Requires fixed infrastructure	Fast set-up and easy deployment
Operational behaviour	Fast and repetitious, mainly task focused	Slow and varied. Focus on task and environment
Footprint/Portability	Large footprint, fixed location	Small footprint, mobile
Purchase cost	Relatively expensive to purchase	Relatively inexpensive to purchase
Profitability	Needs medium to large-volume production	Profitable at low-volume production
Investment prospect	Slow return on investment	Fast return on investment

Traditional industrial robots were designed to perform simple, repetitious tasks (refer to Table 2), typically operating within human protective enclosures [10]. These tasks often form part of a mass production process, where the robot is engaged in the same task for long periods [7]. As such, industrial robots are typically programmed infrequently. In contrast, cobots were designed primarily to assist humans with, among other things, low-volume, customised production [2,11]. Due to the constantly changing task requirements, cobots are typically reprogrammed on a more regular basis [12] as a consequence of their operational flexibility.

Identifying the need for a simplified cobot programming method, Dmytriyev et al. [12] proposed the implementation of a flowchart-based programming environment. El Zaatari et al. [13] found that to increase cobot autonomy, the complexity of the industrial application and the worker's knowledge of the task should be considered when choosing programming features. Schou et al. [14] presented a task-level programming solution using a skill-based system, with the instruction of cobots by operators with no previous robotic experience. The skills, in this context, refer to generic controls related to cobot capabilities rather than operator skills. To narrow the skills gap between robotics experts and workers who lack programming expertise, a system using expert frames, which focus on specific cobot operational aspects, was proposed. One of these frames relates to program quality and identifies syntax errors, unused code and missing parameters or code segments. An interactive interface allows operators to visualise a cobot operation and modify behaviour in response to feedback data received [15].

Understanding the contributing elements would be beneficial to systematically analyse the effectiveness of existing cobot programming and control methods. However, reviews on foundation data, including existing and proposed cobot programming and control methods in the literature, their intended users and the types of programmable cobot tasks, appear deficient. Furthermore, research on the relationship between cobot programming skills requirements and task complexity is lacking. These are the research gap questions this study intends to address:

1. What are the common programming and control methods for cobots in the existing market and literature?
2. What bearing does the complexity of a cobot task have on the skills required to program the cobot to perform that task?
3. How effective are cobot programming and control methods in the existing market and literature? Who is the most appropriate worker to implement the methods, and have the methods been suitably developed for that worker?

2. Technical Readiness for Cobot Deployment

Industry perspectives showed that the most significant cobot adoption hurdle is a deficiency among their staff in the knowledge required to program and interact with a cobot [14,16]. Within the knowledge gap, understanding general cobot technologies, application methods, and practical programming is prominent [17]. Employees with prior knowledge of robot programming are considered a key asset in robotic task allocation, ostensibly due to the wide variety of tasks cobots are capable of performing, such as those listed in Table 3. However, the programming of these tasks was more complex than practitioners expected [18]. This apparent misconception about the ease of cobot programming may well be generated from the marketing promises presented by the cobot vendors in general. This premise will be analysed in Section 5 of this paper. Augmenting the programming skills problem, a fear of programming and technology in general among potential operators has been identified as a possible barrier to cobot adoption [19].

Although there is a significant role for cobots to play in low-volume manufacturing [2,11,20], the time spent programming cobots for small batch runs is often not economically viable [21]. This suggests that developing a simpler and more efficient programming method would significantly increase cobot utilisation, especially within the varied tasks associated with low-volume manufacturing operations.

Traditional industrial robots have typically been programmed by dedicated engineers, who may be located off-site and have limited operational knowledge of the task being performed by the robot [14]. Due to the smaller job runs and requirement for faster and more frequent reprogramming of cobots, it may be more efficient to shift the focus from the skills required to *program* a cobot to the knowledge of the *task* a cobot is carrying out [7]. There seems to be a disparity in programming a cobot to perform a particular task. In most practical scenarios, it is often the person who knows the task well who is not competent with programming, and the person who is competent with programming does not know the task well [22]. A possible solution is that the person with knowledge of the task could also program the cobot [23,24]. Operators with task expertise provide substantial improvements in the precision and efficiency of cobot operations when compared to non-experts [22]. An aim of adaptive task-sharing design principles is that cobot programming should be part of the workers' and assembly planners' duties [25]. Substituting a program engineer for a worker skilled with the task being conducted but without cobot technical skills would require a simplified programming tool.

Table 3. Cobot tasks and related operations (Adapted from [6]).

Cobot Task Category	Related Cobot Operations
Assembly	Screwdriving, Part Insertion, Pick and Place
Dispensing	Gluing, Sealing, Lubricating, Painting, Coating, Dipping
Finishing	Sanding, Polishing
Material Handling	Packaging, Palletising, Bin Picking, Kitting
Material Removal	Grinding, Deburring, Trimming, Milling, Routing, Drilling
Welding	Mig, Soldering
Quality Inspection	Testing, Inspecting, Measuring
Machine Tending	CNC, Injection Moulding, Automated Machining

3. Existing Cobot Programming and Control Methodologies

Depending on the complexity of the task to be undertaken by the cobot, a range of proprietary and third-party solutions are available. For more straightforward cobot tasks, such as the first and second tasks listed in Table 4, teach pendant (also known as *Lead through programming*), teach or program by demonstration, and offline/simulation programming are commonly available options [23,26]. Figure 1 shows the teach pendants produced by ABB and Universal Robots. A variety of other programming options are provided by cobot vendors, as detailed in Table 5.

Figure 1. Teach pendants from ABB (**left**) and Universal Robots (**right**) [Image by authors].

For more complex tasks, such as the last two levels in Table 4, there are proprietary scripts, graphical user interface (GUI) programming applications, and the option of programming in a traditional programming language such as Python, C, C++, C# or Java, or a dedicated robotics language such as Robot Operating System (ROS). ABB's RAPID [27] and Universal Robot's URScript [28] are two typical examples of this category, which require a relatively higher level of programming skill. The collaborative nature of a cobot allows it to perform a greater range of tasks compared with an industrial robot, as shown in Table 2. Many of the programming methods developed for industrial robots could restrict the range of operations and be less adaptable if applied directly to a cobot [29]. Because of this programming rigidity, legacy programming methods may be less intuitive than those developed specifically for cobots.

3.1. Programming of Industrial and Collaborative Robots

Traditional industrial robots have fulfilled manufacturers' automation needs since their introduction in the 1950s [5]. Relieving human workers from repetitious and laborious tasks, these robots have been segregated from humans for safety reasons. Resolutely devoted to its assigned task, the industrial robot blindly follows assigned orders, unable to accommodate sudden changes in circumstances. Requiring relatively infrequent reprogramming, tasks assigned to industrial robots are typically simple, repetitive operations designed to be performed from a fixed location [29]. While some industrial robots are mounted on sliding rails or mobile platforms to increase their mobility, most are stationary because of their bulk and requirement to be housed within protective cages [30]. Their ability to operate at high speed allows them to excel at high-volume manufacturing in the mass production era [31].

A cobot is designed to perform precision tasks at relatively slower speeds while demonstrating high flexibility and interactiveness and allowing a much more diverse range of operational tasks than industrial robots [7]. In summary, both industrial and collaborative robots are practical but have different roles to play in the industry. Directly comparing attributes between the two, such as speed, without considering these different roles [32] can potentially result in misguided conclusions. If applied to cobots, similar programming techniques for industrial robots may not make sufficient allowance for the differences between the two, which are comparably outlined in Table 2.

As modern technology progresses, many manufactured products become more complex, as do their manufacturing processes [33]. Today's intricate manufacturing assemblies and processes are generally beyond the scope of industrial robots, especially where part of the workload must be shared with human workers. With a more compact and streamlined form and the addition of sensors and other safety features, cobots can operate outside protective cages and work safely and collaboratively with humans. This allows a level of automated task flexibility and functional expansion, opening up new opportunities for product diversity in manufacturing [34]. As a result, a much wider variety of low-volume, on-demand product manufacturing is now possible, which is the embodiment of agile manufacturing [7,35]. Accompanying this more flexible and adaptive capability, however, is the need for more frequent reprogramming of cobots [7].

3.2. Cobot Task Complexity

Cobots are one of the most utilised machines within manufacturing automation, capable of implementing a broad range of application tasks [11]. The level of complexity among tasks may vary according to the type of task and the object or workforce being manipulated. For example, a sanding task on an object with a flat surface may be considered less complex than if the object was of a complicated design. Similarly, screwdriving, parts insertion and assembly may be deemed complex tasks due to the fine movements and precision required. Still, the level of complexity may vary depending on the configuration, degree of precision and other specific requirements of that task. Universal Robots has defined a common range of cobot task categories [6], outlined in Table 3. Some examples of tasks for each category are also provided in the table.

Cobots are expected to autonomously manage task complexity, especially as it increases beyond human capacity during collaborative operations [13]. The execution of less complex tasks is often considered an automated process, while for tasks of higher complexities, there is an expectation that they will be conducted autonomously by robots [36]. Furthermore, task complexity can be reduced by sharing the task among multiple cobots [37] or by task reduction, cooperatively solved by multiple robots [38]. For instance, some researchers have explored different ways of coordinating multiple cobots' movements and actions [39], while others have focused on developing new algorithms to improve communication and cooperation among cobots [40]. This paper focuses on the complexity of the task overall to be undertaken by a cobot or cobots with respect to the programming effort required. While multiple-robot systems (MRS) can reduce the overall complexity of a task, their need for collaboration and accurate coordination [41] can add significantly to the programming workload. Further compounding the technical complexity of MRS is the need to select a suitable task allocation strategy that is robust, scalable and can be optimised for a specific task [42]. A cluster-based approach, centred around the location of tasks relative to team members, allocates tasks by training a binary classifier to nominate one of two task allocation mechanisms through an auction bidding process [43]. Another model, using k-means clustering, works to solve the balanced multi-robot task allocation problem by minimising travel distance while optimising utilisation, which relates to task completion time [44]. Intended for use within a Cyber-Physical System such as a warehouse management system, task execution is asynchronously assigned to multiple robots in an ordering mechanism to allocate interdependent Human–Robot Collaboration (HRC) tasks [45]. The model uses mutual exclusion to allocate tasks, dynamically promoting system accuracy and robustness. A *Fuzzy Logic System* can coordinate multiple robots to fulfil a common task. Robots, trained using a *Genetic Fuzzy System*, derive a functional strategy to jointly execute a nominated task [46]. MRS that rely on communication mechanisms, such as decentralised planning algorithms, need complex programs to manage alignment coordination between interacting agents accurately [47].

Adapted from the Universal Robots task definitions, an empirical list of task complexity levels, with associated indicative category examples, is presented in Table 4. The levels have been selected based on the amount of robotic and programming effort required to perform a typical task at the proposed complexity level rather than on the computational resources consumed by a robot during the execution of the task [38]. The degrees of effort have been determined empirically based on a range of programming tests conducted primarily with a UR5e, a cobot representative of the Universal Robots eSeries cobot range [48]. Robotic input refers to the range of manipulator and end-effector actions per task, including:

- Type of joint movement (joint or linear motion)
- Number of movements
- Number of waypoints
- Number of end-effector engagements
- Degree of positional precision required
- Sensor input required (force, proximity, gyroscopic, etc.)
- Ease of end-effector to grip (owing to the surface or shape of parts, etc.)
- Payload handling

Table 4. Task complexity levels.

Task Complexity Level	Types of Associated Tasks
1 (Low)	Simple Gluing, Sealing, Dipping, Sanding, Polishing
2 (Low–medium)	Pick and Place, Lubricating, Painting, Coating, Injection Moulding
3 (Medium)	Material Handling, Simple Assembly, Grinding, Deburring, Trimming, Drilling, Screwdriving
4 (Medium–high)	Parts Insertion, CNC, Automated Machining
5 (High)	Complex Assembly, Precise Milling, Routing, Quality Inspection

Programming input refers to the quantity and complexity of the programming elements required per task, as discussed in Section 5.3. Within the *types of associated tasks*, each task can vary significantly in complexity (see Table 4). For example, while a sanding task conducted on a single flat surface could be considered a task with low complexity, the precision sanding of a spherical object may well place the task at a higher level. It is, therefore, impossible to be definitive with the assignment of the input levels, nor is it the intention within the scope of this study. Instead, the objective is to assign input levels based on the task complexity levels presented, with the *types of associated tasks* being indicative of those levels.

3.3. Task Allocation in Human–Robot Collaboration

A vital step in the optimisation of Human–Robot Collaboration (HRC) is the assignment of tasks to individual collaborators (agents) to maximise the efficient utilisation of the cobot [49]. Within a manufacturing operation, sub-processes can be instinctively allocated to the cobot or the human, based on considerations such as suitability, capability, or human preference. To increase efficiency in a collaborative production environment, a systematic resolution process must be applied to determine which tasks should be conducted by the cobot and which tasks would be best left to a human [50,51]. The individual skillsets of both the cobot and human involved are considered. Tasks are dynamically assigned in a skill-based HRC system, which provides a graphical programming interface and pre-programmed macros to simplify the cobot programming operation [52]. A task allocation model, which maps task characteristics to agent capability, was proposed by [53] and exploited human operators' adaptability and cognitive prowess, along with cobots' efficiency, accuracy, and consistency. Designed to manage tasks allocated to a human and two robots in a heavy part handling HRC assembly operation, a solver based on the *Genetic Algorithm* [54] is used to optimise both operation time and selection based on agent capability. Considering human contentment in HRC, a two-staged capability-based task allocation process was proposed [55]. In the first stage, task elements that align with human capabilities are identified to assess whether humans can perform these safely or are automated to preserve human safety. The remaining tasks are then allocated according to suitability in the second stage based on agent capability, time efficiency, cost, and quality outcomes. Allocation of tasks in HRC assembly operations is based on the classification of the task complexity level and assigned to humans or robots in a complexity-based task allocation method [56]. Agent skills, along with environmental aspects such as component properties, presentation and feeding, are considered in the selection process, which affects efficient handling in an assembly environment. A hierarchical HRC framework proposed by [57] separates task allocation into abstraction and allocation layers in an assembly environment. The abstraction layer defines the planning phase of the collaborative assembly, considering the specific attributes of each element, including human and robot agents and the real-time communication capabilities of each, while the allocation layer manages the skills implementation requirements for the task.

An action scheduling framework using Artificial Intelligence (AI) contains a scheduling algorithm that generates task allocations, which was proposed by [58]. The system considers the sequence of operations, tools required, resource availability, positioning and time efficiency in its optimisation process. Within this human-centric framework, there is a focus on the safety of human operators, and operations are blocked if prerequisite operations have not been completed. Task completion time rewards motivate a *Markov game* model developed by [59], using deep multi-agent reinforcement learning to determine the optimal cobot task allocation schedule. This model evaluates cobot and human agent availabilities and determines the optimal task scheduling policy for the operation within a chessboard structure that minimises all assembly task completion times as its objective.

3.4. Leading Cobot Vendors' Programming and Control Methodologies

To provide readers with an overview of the landscape of existing cobot programming and control methodologies, Table 5 lists seven leading cobot vendors and details of their programming and control solutions. A selection of generic programming applications, which provide an alternative to many proprietary developments, are also included.

Table 5. Programming methods for leading cobot vendors.

Cobot Manufacturer	Programming Interface/Type	Programming Method	Intended User Level	Intended Task Complexity Limit	Source Language/Related Language (If Known)
Universal Robots [28,60,61]	PolyScope	GUI/Hand guide	Unskilled	Low	Proprietary (URP)
	URScript	Textual (Script)	Skilled	High	Python, C++, C#, VB, Java
	URSim	GUI/Simulation	Unskilled	Low	Proprietary (URP)
ABB [62–64]	Wizard	GUI/Hand guide	Unskilled	Low	Block-based RAPID
	FlexPendant	GUI/Simulation	Semi-skilled	Medium	
	RAPID	Textual	Skilled	High	Visual Basic
	RobotStudio	GUI/Simulation	Semi-skilled	Medium to High	Proprietary (RAPID)
Fanuc [65]	TP (Teach Pendant)	GUI/Hand guide	Unskilled	Low	Proprietary (TP)
	Karel	Textual	Skilled	High	Pascal
	Roboguide	GUI/Simulation	Semi-skilled	Medium to High	
Festo [66]	BionicCobot	GUI	Unskilled	Low to Medium	ROS
	Festo Robotic Suite	GUI	Semi-skilled	Medium	Python/ROS
	RoboCIM	GUI/Simulation	Semi-skilled	Medium	Proprietary
Kuka [67]	iiOKA OS	GUI/Hand guide	Unskilled	Low	Linux kernel
	KUKA Work.Visual	GUI/Textual	Skilled	High	
	KRL (KUKA Robot Language)	GUI/Textual	Skilled	High	Pascal
	Kuka Sunrise	Textual	Skilled	High	Java
	Kuka Sim/SimPro	GUI/Simulation	Semi-skilled	Medium to High	Proprietary
Yaskawa [68–70]	Direct Teach (DT)	Hand guide	Unskilled	Low	Proprietary
	Smart Pendant	GUI/Hand guide	Unskilled	Low	Proprietary
	INFORM II	GUI/Textual	Skilled	High	C
Omron [71]	ACE (Automatic Control Environment)	GUI/Hand guide	Unskilled	Low	C#
	eV+	Textual	Skilled	High	MS-DOS/Unix Script
	ACE Emulation Mode	GUI/Simulation	Unskilled	Low	C#
Generic cobot programming applications (RoboDK, ArtiMinds, Wandelbot, Pickit, Robomaster, G-Code, ROS, Traditional languages)	RoboDK GUI [72]	GUI/Hand guide	Unskilled	Low	RDK
	RoboDK API [72]	Textual	Skilled	Medium–High	Python (default), C, C++, . . .
	ArtiMinds RPS [73]	Modular	Semi-skilled	Medium–High	Proprietary (RPS)
	ArtiMinds RPS [73]	Modular	Semi-skilled	Medium–High	Proprietary (RPS)
	Wandelbot Tracepen [74]	Input device/App	Unskilled	Medium–High	Proprietary
	Pickit robot vision system [75]	GUI/3D vision	Semi-skilled	Medium	
	Robotmaster [76]	GUI/Simulation	Semi-skilled	Medium–High	
	G-Code [77]	(CAD/CAM)/Textual	Skilled (CNC)	Medium–High	G-Code
	ROS/ROS2 [78]	Textual	Skilled	High	C++/Python
	Traditional prog languages: Python, C, C++, C#, Java, [79]	Textual	Skilled	High	Proprietary

The *'Intended User Level'* and *'Intended Task Complexity Limit'* fields in Table 5 are broadly based on cobot vendors' statements, including those in Section 4.1 of this paper, and practical reviews of the various applications. The aforementioned fields will be the subject of the comparative analysis presented in Section 6. In addition, *Intended User Level* references the general skill level definitions stated in [80], with unskilled work requiring little to no independent judgement or previous experience to perform simple tasks; semi-skilled may require close attention and coordination abilities for tasks with some complexity, with decisions made by others; and skilled work, which requires judgement and decision-making abilities while performing more complex tasks.

3.5. Controlling Cobots without Formal Programming

Recognising the need for frequent reprogramming and flexible, collaborative operation, cobot vendors compete to find the most intuitive and efficient method of controlling their cobots.

3.5.1. Existing Cobot Control Methods

A common easy-to-navigate control method offered by cobot vendors is 'Guiding' (sometimes referred to as the Teach method/Easy programming/Basic programming), which involves human manipulation to 'teach' the cobot the required sequence of operation. Typically, user interfaces combine a pendant, tablet or PC-based display with the user positionally hand guiding the cobot, as shown in Figure 2.

Figure 2. Programming UR5e with the teach pendant and hand guiding by a human operator.

3.5.2. Proposed Cobot Control Methods in Recent Literature

The benefits of assigning cobots to small-batch agile production [81] can be eroded by the requirement for more frequent reprogramming [7], especially if there is a need to continuously hire skilled programmers [12]. However, by simplifying programming and control methods sufficiently, workplace operators, rather than dedicated programmers, could intuitively program the cobots [82]. In this case, there would be greater potential for increased cobot utilisation since the programming role would be open to a broader range of workers.

Skill Based System

Since cobots are designed to interact closely with human workers, who have extensive knowledge of the task being undertaken, programming the cobots by the same workers would seem to be a logical part of the collaboration. With an emphasis on the task rather than the program, a Skill Based System (SBS) allows a robotic novice worker to use a graphical interface to map skills relating to a task, which equates to individual robotic actions. As shown in Figure 3, these skill entities are used as parameters to initiate the kinaesthetic configuration of a cobot's joints [14]. For some prospective programmers, however, the first step will be to overcome their fears of programming and technology and accept cobots as collaborative partners rather than their future replacement [19].

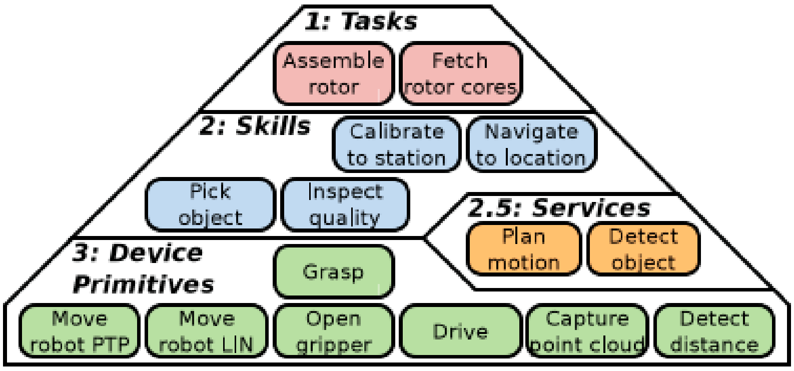


Figure 3. An example of a skills-based system with 3 abstraction layers is used to configure cobot joint settings [14] (by permission).

Block-Based Interface

One approach in the quest for a more comprehensible programming experience is a block-based interface, where pre-programmed modules can be selected and placed in a desired sequence by novice users, as shown in Figure 4 [83]. This type of drag-and-drop interface has two key advantages over text-based programs. Firstly, the blocks, written in simple language, are designed to be intuitively arranged by a novice while providing visual cues as the program develops. This allows the user to focus on the task's workflow rather than the program's syntax, which relates to the second advantage. Each block can be considered a pre-programmed function, which is syntactically correct and, along with the other blocks in a sequence, collectively compilable [84].

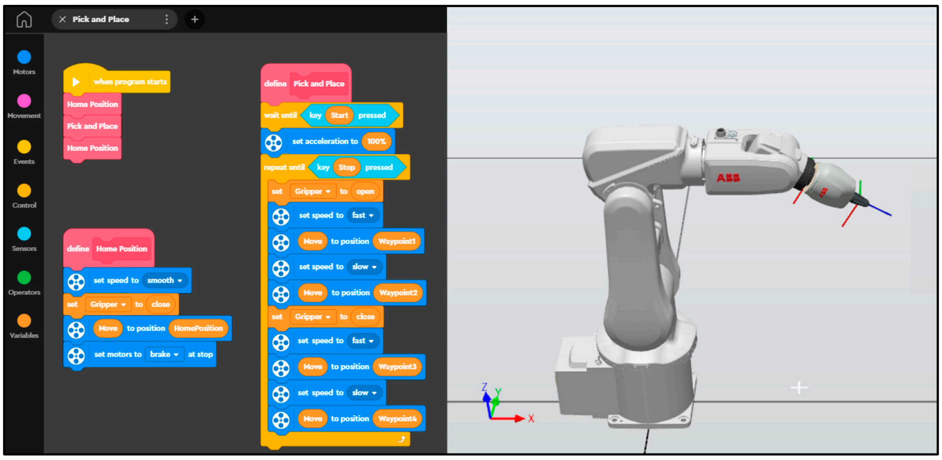


Figure 4. Block-based programming editor with a virtual industrial robot (Adapted from [83]).

Layered Block-Based Interface

A layered approach to block-based programming provides users with different skill levels and the flexibility to carry out programming tasks with a complexity level that matches their ability. From workers with little technical skill generating basic assembly workflows to those experienced in programming and robotics undertaking more complex programming tasks, a three-layered approach caters for a broad range of user abilities and task complexities in a single system (refer to Table 6) [24].

Table 6. A three-layer block-based cobot programming model [24] (by permission).

Programming Layer	Roles and expertise	Required Training	Support Techniques
Layer 1: Basic assembly workflows (robot movements and tool actuation)	Assembly workers and laypersons with some assembly experience	Some technical training (e.g., professional school)	Mounting and assembly devices, multimodal teach-in
Layer 2: Block-based programs (task blocks, variables and control structures)	Industrial engineers with computational thinking abilities and technical intuition	Formal technical training and a programming course	CAD-modelling, 3D printing, laser cutting
Layer 3: Advanced functionality (databases, connectivity, etc.)	Software engineers with advanced programming skills	Formal software engineering training	Internet/intranet, databases, cloud, Manufacturing Execution System (MES)

Chat-Assisted Block-Based Interface

By enhancing block-based programming with natural language chat functionalities, non-technical users can more intuitively assign programmed tasks to cobots. Furthermore, this flexible hybrid approach can allow users to implement more complex tasks more easily than block-based programming alone while building user acceptance of the technology and confidence with cobot programming over time [85].

Artificial Intelligence

Artificial Intelligence (AI) can also assist a programmer with specific language code suggestions offered in real-time, generated through the Natural Language Processing (NLP) functionality of OpenAI Codex [86]. However, programming expertise would be required for each suggestion as part of the accept/reject/edit decision-making process. Developing autonomous cobots motivates many researchers, and they are exploring specific areas of AI that are more suited to efficient cobot management. For instance, *Reinforcement Learning* has been effectively used to train robots to conduct complex tasks; while modifying their behaviour in response to changes in their environments, which they continuously monitor. In contrast, Deep Learning (DL) could restrict dynamic response behaviour due to time-inefficient data processing requirements [87]. According to [87], DL may not be suitable for differentiating objects of similar appearance, which may be the case with some electronic components used in assembly processes. In addition, DL is a complex area of AI that requires significant experience and skills to implement [88], a notion that opposes the objective of the current study. To improve cobot adaptivity and intelligence in HRC, an AI-based 3D perception system, which incorporates *RGB* colour scan cameras to generate 3D representations of a cobot's operational environment, contains an anti-collision function and allows a human operator to control cobot movement with natural gestures [89]. Furthermore, a path planning architecture establishes a predetermined path for *Unmanned Aerial Vehicle* robots, using the *Hungarian algorithm* to optimise cost efficiencies. A 3D occupancy map of the cobot's operational environment is generated to ensure a collision-free path [90].

Voice and Gesture Control

A more instinctive cobot control measure that alleviates the need to formally program allows a user to guide a cobot through a task using voice commands and physical gestures. An image and sound processor comprises a camera, a depth sensor (for human movement tracking) and a set of microphones that capture the relevant human speech and gesture inputs. These are converted in the software program into corresponding coordinates for the cobot joints [91]. Figure 5 shows the testing station for the speech and gesture system, depicting the operator, graphical interface and ABB IRB120 robot.

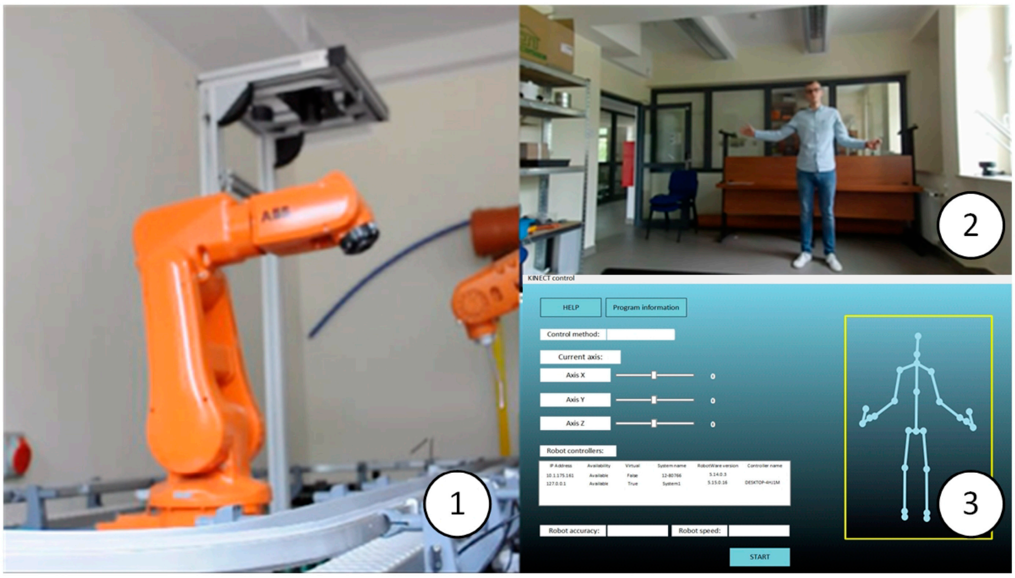


Figure 5. A testing platform for cobot voice and gesture control system, proposed by [91], 1. ABB IRB120 robot, 2. the operator, 3. graphical interface (by permission).

A validation process is used between gestures in a task-managed system to prevent undesired sequences from executing. In their work, users can decide to return to a previous save point of the program by way of a predefined human gesture [92]. In these systems, operators would need to be suitably trained on the range of valid speech and gesture commands.

Virtual Reality Systems

Moving further away from traditional programming and closer to a gaming platform, there is an immersive cobot control method. Controlling a physical cobot from a virtual environment is a concept that replaces a standard cobot interface console with a virtual reality (VR) system, which is connected to the cobot's controller mechanism. From within the virtual environment, a human can control a virtual cobot (a virtual representation of a physical cobot), with the inputs stored in a dedicated database so that the cobot's trajectories can be visualised in real-time or reproduced from stored values [93]. In addition, cobot control includes safety precautions in human–cobot interaction, which can be achieved in a VR environment. Cobot behaviour is dynamically modified to anticipate human movement using a motion tracker [94].

Augmented Reality Systems

Combining the virtual environment with the real world, augmented reality (AR) systems can provide a more realistic human experience than VR systems. AR systems allow a faster and more user-friendly approach for humans to interact with cobots, with users preferring an AR system over traditional teaching pendants or console options [95]. Users with no previous programming experience can perform a range of cobot motion tasks through an AR interface. However, as the complexity of the cobot task increases, so does the difficulty experienced by the user. For example, pick and place tasks are more easily conducted than welding tasks. Overall, the AR experience positively affects the novice user's confidence to accurately and safely carry out the assigned tasks [96].

4. The Cobot Programming and Control Sales Pitch

With the global cobot market expected to exceed USD 9 billion by 2024 [9], competition is fierce among cobot vendors as they vie for dominance in this lucrative industry. One of the cobot vendors' prime objectives is to simplify the human–cobot interaction method.

Furthermore, to alleviate the need for robotic or formal programming skills, cobot vendors attempt to present users with the most intuitive interface to control the cobot.

4.1. Cobot Vendors' Perspectives on Their Programming Applications

In this section, a selection of claims and statements in the leading cobot vendors' product description material has been quoted on the program and control attributes of their cobot interfaces aimed at untrained users (refer to Table 7). The claims made by the cobot vendors will be used to estimate programming effectiveness and intuitiveness from the perspective of the intended user.

Table 7. Cobot vendors' marketing claims.

1. Universal Robots (PolyScope) [60]
"Use your process expertise and PolyScope's graphical interface to create a robust automation system. No Code? No Problem." "PolyScope connects operators to robots for efficient and productive automation. You don't need coding experience to automate your processes." "Build programs by selecting nodes from a menu and placing them in order of operation. Each node represents an instruction for the robot and its parameters can be configured."
2. ABB (Wizard) [62]
"Wizard easy programming—An easy and intuitive way to program cobots and Industrial robots." "Program your robot application within minutes! Wizard is an easy graphical programming interface for ABB Cobots." "With Wizard, anyone and everyone can program their robot application." "Only a few minutes after the installation you will be able to operate your robot. With Wizard's easy drag and drop blockly based programming software, no specialized training or programming skills are required."
3. Fanuc (TP) [65]
"But the CRX's #1 "Ease of Use" benefit is its Simple Drag and Drop Programming. 30 years ago, robot programming was a high-level structured language like Fortran or C++. An engineer or maintenance technician went to school to learn how to learn the programming language. Both are very powerful, but not easy to use. With the all new FANUC CRX Tablet Teach Pendant–simple programming becomes reality. Easily program and teach points with the CRX Tablet Teach Pendant. The drag and drop interface for lead-through teaching and simple programming is easy with no prior robotic experience needed."
4. Festo (Festo Robotic Suite for BionicCobot) [66]
"Programming a robot is child's play" "The BionicCobot is operated intuitively via a graphic user interface developed in-house." "Commissioning and programming are intuitive, quick and easy with the "Robotic Suite" software." "The developers were focusing specifically on making the Robotic Suite, the actual heart of the cobot, as simple to operate as possible so there is no need for prior programming knowledge." "When it comes to programming, most of us probably think about complicated lines of code with lots of abbreviations, brackets and other symbols. But programming a robot can actually be very easy, as is shown by the software that Festo developed for its pneumatic lightweight robot, the BionicCobot."
5. Kuka (iiQKA OS) [67]
"iiQKA allows you to put together your individual automation package, without any prior knowledge or programming experience." "The intuitive graphical interface allows for fully autonomous control of the system without any programming knowledge." "Designed for quick start-up with little to no expertise, iiQKA offers incredible speed to integrate robots."

Table 7. *Cont.*

6. Yaskawa (DT) [68,69]
"Quick and easy programming." "Simple and intuitive operator control, short learning curve." "Ideal for users who need to carry out frequent reprogramming and thus appreciate simple operator control." "Ideal for novice robot users, this pendant simplifies INFORM programming for easy-to-understand operation and fast implementation of the robot system." "The perfect entry into programming. Simply move the robot flange by hand, record the motion points and operate the gripper actuation by pressing the respective DT buttons. The code is automatically generated in the background on your pendant."
7. Omcron (ACE) [71]
"The Automation Control Environment (ACE) software allows you to build applications, such as Pack Manager packaging applications, which can be basic pick-and-place cells or complex cells with multiple cameras, conveyors, and robots. You can create and configure these cells without having to write any programming code. For applications that require greater control, you can override the default V+ program code and make changes as needed."

4.2. Epitomising the Marketing Message

Regarding user interaction with their products, the general message from cobot vendors is that neither robotics nor programming experience is required, at least for the less complicated cobot tasks. However, the complexity level is not explicitly referred to in their marketing blurbs. Furthermore, many instructional videos released by cobot vendors [97,98] tend to emphasise the simplicity of their user interfaces, while limitations, in terms of task complexity, have generally been concealed. Marketing cobots as "simple to program" can lead to the assumption that skilled programmers are no longer required [20]. This assumption makes no allowance for the complexity of the task to be programmed, nor does it consider the variation in the skillset required to do so. Despite cobot vendors' assurances that unskilled personnel can program their cobots, others have a different perspective. A leading cobot tool manufacturer views cobot programming as a job for an engineer with mechanical, electronics, electrical and programming skills and an understanding of combined system functionality and associated theory [99]. Manufacturers in the industry, who operated cobots, revealed that of the total time employees spent with their cobots, automation engineers accounted for 48%. Approximately half of the automation engineers' cobot duties were dedicated to programming [100]. In Section 6, the cobot vendors' claims will be compared with the observed skills required to program a cobot, to establish the validity of their claims and the effectiveness and suitability of the programs for their intended purposes and at their intended user levels.

5. The Underlying Skillset

There are three key elements involved in the process of cobot task allocation [101].

I. The cobot and its environment
II. Cobot programming steps
III. Cobot task implementation

For a human tasked with implementing the process, these elements correspond to the knowledge and skills that must be acquired or provided extrinsically. Regarding cobots, a conceptual knowledge of the hardware and technical skills in cobot operation and problem-solving would be necessary, along with knowledge of the assembly line or other cobot environments. To program cobots, knowledge of the programming application is essential, along with skills in programming within the context and capabilities of a specific cobot. Finally, knowledge of the task the cobot is to perform, optimisation characteristics, and the skills to carry out the steps of the task would typically be required [101]. An aim of adaptive task-sharing design principles is that cobot programming should be part of the workers' and assembly planners' duties [25]. Substituting a program engineer for a worker skilled with the task being conducted, but without cobot technical skills, would require a simplified programming tool.

5.1. Fundamental Technical Aspects of Cobot Control

To highlight the complexities potentially experienced by the user of a typical cobot programming interface, robotics and programming concepts and keywords are presented. Table 8 relates to the robotic (hardware) elements, while Table 9 contains the programming (software) elements. The reference source for both tables was the Universal Robots *PolyScope* graphical interface [60], used to program a UR *e*Series cobot [102]. The list is not intended to be exhaustive but rather a selection of representative fields more relevant to the programming than the setting up of a cobot.

It is acknowledged that in Tables 8 and 9, not every listed entity would need to be configured for each program. However, it could be argued that in support of sound judgement, a knowledge of the purpose and functionality of each would be required to determine when they *should* or *should not* be configured. For example, there is a view in the cobot tool manufacturing industry that cobot programmers should have adequate knowledge of advanced programming functions [99], such as those listed in Table 9.

5.2. Robotics (Hardware)

A selection of robotic control elements, along with corresponding robotic entities, is listed in Table 8.

Table 8. Robotic control elements.

Robotic Element	Specific Robotic Entity
General	Fundamental kinematics and dynamics principles
Limits	Safety limits, including power, momentum, stopping time and distance, tool speed and force and elbow speed and force limits
	Joint position range limits and maximum speeds
Orientation and positioning	Tool Centre Point (TCP), Tool Offset, Tool Position and Tool Rotational Vector settings (all represented as three-dimensional cartesian coordinate frames)
	Relationship between base and tool coordinate frames
	Direction for linear movement (expressed as positive or negative cartesian coordinates or direction vector)
	Waypoints (with options of fixed, relative or variable position)
Joints	Joint positions (in degrees) for the base, shoulder, elbow and three wrist joints (pitch, yaw and roll).
	Linear, non-linear and circular joint movements
	Joint speed and acceleration values
Communication	I/O Signals (Digital, Analog, Tool, Configurable, Boolean Register, Integer Register, Float Register)

5.3. Programming (Software)

A selection of programming elements, along with corresponding cobot programming entities, are displayed in Table 9.

5.4. Skills Required to Program and Control a Cobot

An empirical list of the skills required to program and control a cobot has been created to analyse the relationship between these skills and the complexity of the tasks they are used to conduct. Descriptions have been provided for each pair to clarify the assignment of skill values and levels. TACOM [103], a *ta*sk *com*plexity measure, was used to guide the selection of cobot task complexity categories. The metric is based on the quantity of information an operator must process about the task, the number of actions and logical sequence a task contains, current knowledge of the task and available cognitive resources for decision-making. TACOM calculates task complexity with respect to the task's procedural steps. The information density and composition that define a task and the number of actions and order of operation required to execute it affect the complexity level of a task. A certain amount of system knowledge is required to carry out an action and to understand the complications of the task. The capacity of an operator, in terms of the precision and cognitive effort, during the execution of a task, along with the specific resources required by the task, contribute to the performance level of a human operator [104].

Table 9. Cobot programming requirements.

Programming Element	Specific Cobot Programming Entity
General syntax	Program structure and sequencing
Data	Constants, variables, variable assignment
Conditionals/selections	If, ElseIf, Else, Until, Switch statements, Boolean
Program loops/iterations	For, While, Do-While
Functions/procedures	Thread, Subroutine call
Control points	Event, Wait, Set, Halt, Timer

Table 10 outlines a proposed user skill level paradigm. Each skill level designation has a range from a low to a high skill level listed, followed by a suffix, which indicates the skill type (R = Robotics, P = Programming). For example, '1 R' indicates a skill level on the low end of basic robotics knowledge, while '3 R' refers to the high end of the same level. There is a slightly broader range (7–10) in the higher levels of both skill types to accommodate the greater range of advanced technical concepts.

Table 10. User skill levels for the programming and control of cobots.

Skill Level Designation	Description of Skill Level
0: Unskilled	No knowledge of robotic operation or programming
1–3 R: Basic Robotics	Basic knowledge of robotic concepts, such as the difference between collaborative and industrial robots, the joint structure of a manipulator (robotic arm) and basic understanding of end effectors such as a gripper or suction cup
4–6 R: Mid Robotics	Familiar with robotic movement and functionality, including linear and non-linear joint movement, consequences of joint speed and acceleration settings (collision prevention) and coordinate frames
7–10 R: High Robotics	In-depth knowledge of robotics, with practical skills in cobot installation, tool configuration with respect to coordinate frames, precise joint configuration, I/O signals, sensors and configuration of safety elements such as protection zones
1–3 P: Basic Programming	Basic knowledge of programming concepts, such as data types, data inputs, computations and outputs
4–6 P: Mid Programming	Familiar with basic programming techniques involving common elements such as variables, loops and conditionals
7–10 P: High Programming	Competent in structured programming, using functions, different loop and conditional types, switch statements and classes

Although cobot programming methods range from those that are easier to use, for less complex tasks, to those requiring higher skills for more complex tasks, there is some common ground between them. Following some targeted programming comparisons, it was found that some tasks of medium complexity could be programmed at the higher end of a Teach Pendant's capability, for example, or at the lower end of a script-based method, as indicated with the Venn diagram in Figure 6. The choice of method would be subject to the skill level or discretion of the person programming the task. This further complicates mapping task complexity to the corresponding required skill level for tasks within that intersecting zone.

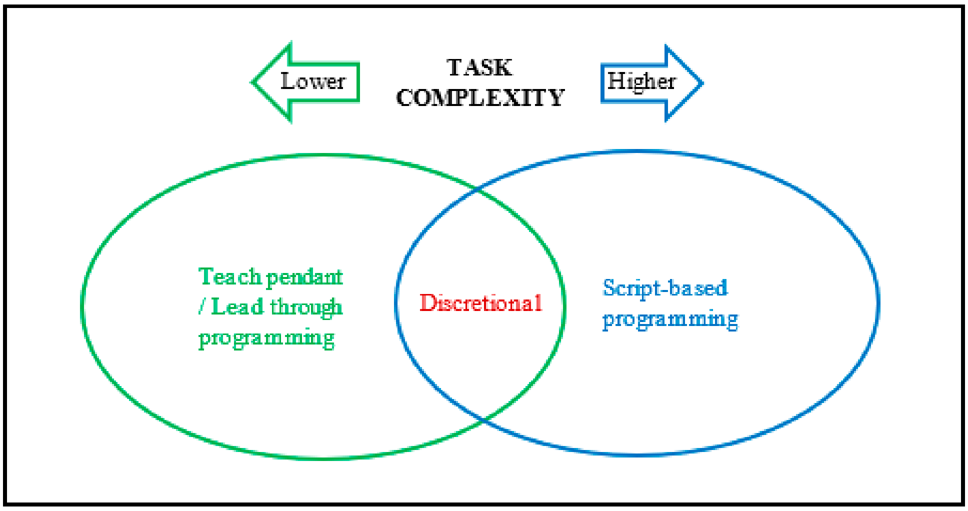


Figure 6. Cobot programming methods according to task complexity.

The extent of the skills and knowledge required will vary based on the complexity of the task being programmed. In addition, the broad range of approaches available to program a specific task can further complicate the job. For example, the kinematic flexibility of a cobot, while allowing great freedom in tool orientation and position, also increases the programming complexity [93]. Considering the robotics and programming skills required, Figure 7 shows a graphical representation of the skills required to program a cobot by different methods based on task complexity.

From a skill requirement perspective, the transition from programming a cobot with a teach pendant to write a script-based program to do so is significant. Using the Universal Robots *URScript* or ABB *RAPID* programming methods as examples, a programmer must

1. Establish a connection to the cobot controller from a remote console. Some knowledge of computer networks would be required to communicate with a host over a socket connection.
2. Compose a syntactically correct control program. At least a moderate level of skill in programming, with an understanding of program structure and syntax, would be required. For this study, *URScript* test programs were written in *Python*, so an understanding of the relevant formal language for the client program is also necessary.
3. Choose:

- Individual cobot joint positional coordinates and orientation parameters
- Motor speed and acceleration settings
- Tool selection
- Delay timing
- Other aspects depend on the complexity of the task being programmed.

Sound knowledge of robotic functionality, particularly about coordinate frames and joint movement, would be mandatory.

5.5. Skills Versus Task Complexity Testing Methodology and Results

A UR5e cobot was programmed to simulate representative tasks from the associated task categories outlined in Table 4. Four tasks with complexity levels ranging from 1 to 8 were programmed with the *Teach Pendant*, while two tasks with complexity levels ranging from 7 to 10 were programmed in *Python* through *URScript*, for comparison. The range of technical skills utilised during the configuration of each programmed task was recorded. These related to the robotic and programming elements, as defined in Tables 8 and 9, respectively, are used to describe the user skill levels outlined in Table 10. The results of the recorded program tasks are shown in Table 11 and represented graphically in Figure 7. The programmed tasks listed in Table 11 were selected to represent each of the task complexity levels in Table 4. The aggregate skill levels in the table are the product of the corresponding robotic and programming skill levels defined in Table 10.

Table 11. Task complexity versus skill level.

Task Complexity Level	Programmed Task	Programming Method	Robotic Skill Level	Programming Skill Level	Aggregate Skill Level
1–2 (Low)	Simple Polishing	Teach Pendant	2	1	2
2–4 (Low–medium)	Simple Pick and Place	Teach Pendant	4	2	8
5–6 (Medium)	Simple Assembly	Teach Pendant	6	4	24
7–8 (Medium–high)	Parts Insertion	Teach Pendant	6	4	24
7–8 (Medium–high)	Parts Insertion	Script-based	9	9	81
9–10 (High)	Complex Assembly	Script-based	10	10	100

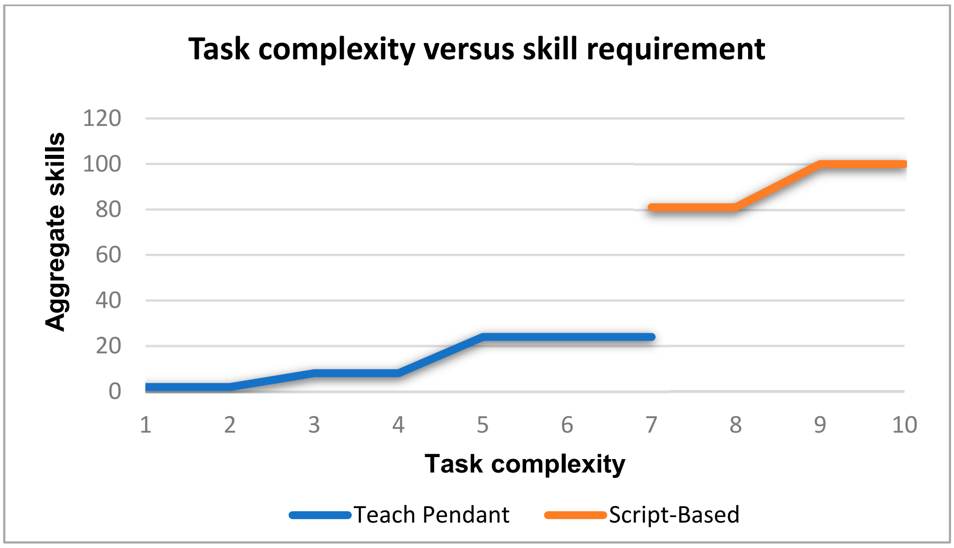


Figure 7. Robotics and programming skills required versus task complexity.

The effectiveness of the teach pendant as a programming tool is demonstrated in Figure 7, with a low level of skills required to program cobot tasks in the lower complexity range. However, from the medium task complexity level, the required skill level is three times higher than the previous one. When using the teach pendant, the required robotics skills are 1.5 times higher than programming skills for all complexity levels. This is because of the greater precision and finer tool positioning typically associated with more complex tasks. At the same time, there is a more gradual increase in skills required with the simplified graphical programming interface of the teach pendant. As task complexity enters the medium to high band, for tasks beyond the capability of the teach pendant, the required skills are 3.3 times higher for script-based programming. Because of the need to precisely define joint coordinates and other cobot parameters within the syntax of a textual programming language, the required robotic and programming skills increase at the same rate. The lack of skills issue has imposed constraints on the programming of complex tasks.

6. Analysis of the Findings of This Review

An analysis was undertaken into the relationship between the complexity of the task to be executed by the cobot and the relative skill levels required of the programmer. Moreover, a mapping of this relationship was established. In the quest to find the most suitable candidate for the cobot programming role, consideration was given to the frequency of cobot reprogramming, knowledge of the task to be performed by the cobot, economic efficiency and organisational logistics. Finally, the existing cobot programming and control methods and those proposed in the literature were evaluated to determine if they were appropriate for the person with the programming role or whether another solution was needed. Section 6.1 evaluates the effectiveness of existing cobot programming and control methodologies, while Section 6.2 presents the practicalities of the main cobot programming and control proposals in the literature. A summary of the findings, including responses to the research questions, is provided in Section 6.3.

6.1. Effectiveness of Existing Cobot Program and Control Methodologies

Cobot tasks programmed with script-based methods require some expertise in formal programming, an understanding of the specific cobot functionality and a sound knowledge of robotics in general. Proprietary Teach Pendants, however, are typically marketed as user-friendly tools designed for programming a cobot without the need for robotic or programming skills. While the graphical interface of the teach pendant is easier to use than a script-based alternative, this study has found that some foundation skills in robotics and programming are still required. Basic programming and robotics knowledge will contribute to a more accurate and safer outcome, even when programming a relatively simple cobot task. Knowledge of coordinate frames, for example, can allow more positional precision of joints and an understanding of efficient joint and linear movements, along with motor speed and acceleration settings, could help with collision avoidance. Furthermore, programming fundamentals such as sequences and loops can add vital insights into program structure and process iteration. When marketing their teach pendants, cobot vendors tend to highlight the ease of use for non-experts without emphasising the extent to which task complexity may complicate the process. The teach pendant, therefore, does require both robotic and programming skills, which increase as the task becomes more complex. It is, however, a significantly more intuitive cobot programming tool than a script-based language, which requires a high level of competency in robotics and formal programming. Some computer networking knowledge may also be required, depending on the method of connection between a terminal and its host.

6.2. The Practicality of the Primary Cobot Program and Control Methodologies Proposed in the Literature

The main cobot programming and control methods proposed in recent literature and reviewed in Section 3.5.2 are critiqued in this section.

6.2.1. Block-Based Interface

While alleviating syntax errors, programming with a block-based graphical interface [83] requires some knowledge of program flow control and an understanding of robotic movement when using it to control a cobot. There is also the potential for runtime errors if, for example, unattainable values have been entered as joint position, speed, or acceleration parameters.

6.2.2. Voice and Gesture Control

Although they present an intuitive approach to controlling cobots, gesture and voice control have their limitations. Gestures can be misinterpreted by the image processor or incorrectly posed by the human operator. An example of where such errors could occur is in the subtle pose differences between *'Axis Y move forward'* and *'Axis Z move forward', where only a slight difference in left arm position separates the two commands*. Voice control is affected by noise variation and also human error in the case of an incorrect command or poor diction [91], which could be challenging in a production environment. Collectively, this method may be difficult for a human to orchestrate, having to coordinate the correct gestures and voice commands, especially when sequencing the cobot through a complex task. In addition, task parameterising is time-consuming, some of the more difficult gestures to pose cause fatigue and while the gesture recognition rate is generally high, instances of false positives and false negatives occur, of which the user is notified, following a validation process [92].

6.2.3. Virtual and Augmented Reality Systems

Work in virtual and augmented environments is progressing; however, VR [93] and AR [95,96] systems for cobot operations do not currently appear to be developed and tested to the point of practical deployment for the industry. What has been proposed in the literature are VR systems that allow users to simulate robotic movement in an immersive environment. However, each use case often requires creating a new virtual environment and current capability is limited to simulation rather than functioning as a practical programming method [93]. AR systems produce tracking inaccuracies, which can misdirect the cobot, and there are also visual constraints. A reduced field of view from the AR wearable devices and occlusion problems restrict the user's perspective of the computer-generated content and could impose risks when deployed in workplaces. In addition, users can be distracted as they continuously swap between augmented and real environments, disrupting the AR system's fluidity [95].

6.3. Summary of Findings and Responses to Research Questions

Section 3 of this paper describes the common programming and control methods for cobots, existing and proposed in the literature. Solutions from seven leading cobot vendors were evaluated, and systems proposed in the literature were critiqued. The impact of a cobot's task complexity on the skills required to program it was addressed in Section 5, with a graphical representation and summary of the analysis provided at the end of that section. The effectiveness of existing cobot programming and control methods and the practicality of methods proposed in the literature were assessed in Sections 6.1 and 6.2, which focused on existing and proposed programming and control methods, respectively. The final considerations were to determine the most appropriate candidate for the role of cobot programmer and assess whether the current or proposed solutions can be matched to that person's skillset. In Section 2, it was argued that rather than focusing on the skills required to *program* a cobot, the knowledge of the *task* should be the focal point. The view was that it is often the case that the person who knows the task well, is not competent with programming and the person who is competent with programming does not know the task well. Operators with knowledge of and experience with a task that a cobot is intended to assist with possess valuable insights into precision and efficiency, details of which may be difficult to relay to a contracted programmer. Moreover, adaptive task-sharing principles aim to embed cobot programming into workers' duties. Based on these findings, a task-experienced worker would be the most appropriate programmer, but the programming methods analysed have not been specifically developed for a novice, as discussed in Section 5. Compounding this problem, the scale in Figure 7 indicates that existing cobot programming and control methods require more skill as cobot task complexity increases. In addition, the investigation in Section 6.2 revealed that the methods proposed in the literature were impractical or not sufficiently developed for reliable deployment. In general, these current cobot programming and control methods and those proposed in the literature require more advanced skills or are underdeveloped to be used by a worker with no robotic or programming experience. In the absence of a designated cobot programmer, therefore, either the upskilling of cobot novices, who are task-savvy operators or the development of a new generation of simplified cobot programming and control methods are the only viable options.

7. Discussion and Future Work

Several elements restrict the establishment of a complete, accurate, and clearly defined cobot task complexity to skill requirement matrix. Cobot task complexity is difficult to define due to variations in the definers' perceptions of *complexity*, which in turn, is conceivably shaped by their levels of skill and approaches to the programming of the task. For example, someone with more programming expertise may see a task as less complex, which could seem daunting to a less skilled programmer. Another key issue is that of the programming method used, especially concerning the tentative selection area between programming methods of different complexity, as discussed in Section 5.4 and summarised in Figure 6. Within that discretionary zone, a task may require more skills if programmed with a script-based method and less with a Teach Pendant.

The ramifications of an incorrectly programmed cobot can be severe, regardless of the programmer's skill level, although programming errors might be more likely with a less skilled programmer. Consequences of a poorly programmed cobot could range from a delay in a manufacturing process, damage to or destruction of the cobot, equipment or products, to human injury or even fatality, particularly if manipulator speed or force settings are not constrained. In addition, companies expose themselves to possible legal action due to injuries caused by their robots, with significant financial claims filed, especially if negligence is a factor [105].

The purpose of this paper was to review cobot programming and control methods and present a broad view of the complexity of the methods and cobot tasks from the perspective of a user skill level. Considerably more research and analysis should be conducted in cobot programming and control methods, focusing on flexibility and ease of use. Humans are the principal collaborators with cobots, so there should be a close connection to the human skills required to interact with a cobot partner. With further development, existing, proposed or hybrid systems could lead to a new generation of human-centric cobot smart control.

8. Conclusions

In a collaborative environment, there is an interaction between humans, who know the task, and cobots, who have been programmed to perform their part. Such an environment consists of many variables, from the type of cobot and programming method used and the programmer's skillset to the complexity of the task. Task and program complexity, along with programming skill levels, have been considered in broad terms and compared with existing and proposed programming and control methods in the literature to evaluate the relationship between task complexity and the skills required to program cobots. This analysis was then used to assess the claims made by the leading cobot vendors about the skills required to program their cobots. Cobot vendors typically emphasise the ease of programming their cobots without reference to task complexity. The findings of this study have revealed that even for tasks of relatively low complexity, some level of robotic and programming skill is required to ensure a safe and effective outcome. As task complexity increases, so do the required robotics and programming skills, contributing to a skills dilemma. Furthermore, to complete highly complex tasks, the programming skills required often exceeded those of the robotics skills. A task-focused approach highlights the benefit of the cobot programming role being performed by the worker with the best knowledge of the task rather than an expert with the best programming knowledge. The worker's lack of programming experience adds to the skills dilemma. Existing and proposed solutions are not currently suitable for that type of deployment. The solution is a programming system requiring no technical expertise, regardless of the task complexity level.

Author Contributions: Conceptualization, P.G., C.-T.C. and T.Y.P.; methodology, P.G.; validation, P.G.; formal analysis, P.G.; investigation, P.G.; resources, C.-T.C. and T.Y.P.; writing—original draft preparation, P.G., C.-T.C. and T.Y.P.; writing—review and editing, P.G., C.-T.C., T.Y.P. and K.N.; visualisation, P.G.; supervision, C.-T.C., T.Y.P. and K.N.; project administration, C.-T.C. T.Y.P. and K.N.; funding acquisition, C.-T.C. and T.Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to acknowledge the support of RMIT University for providing resources and facilities to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bloss, R. Collaborative robots are rapidly providing major improvements in productivity, safety, programming ease, portability and cost while addressing many new applications. *Ind. Robot. Int. J.* **2016**, *43*, 463–468. [CrossRef]
2. De Backer, K.; DeStefano, T.; Menon, C.; Suh, J.R. Industrial robotics and the global organisation of production. *OECD Sci. Technol. Ind. Work. Pap.* **2018**, *3*, 1–44.
3. Benotsmane, R.; Kovács, G.; Dudás, L. Economic, social impacts and operation of smart factories in Industry 4.0 focusing on simulation and artificial intelligence of collaborating robots. *Soc. Sci.* **2019**, *8*, 143. [CrossRef]
4. Bryndin, E. Collaboration Robots with Artificial Intelligence (AI) as digital doubles of person for communication in public life and space. *Bp. Int. Res. Exact Sci.* **2019**, *1*, 1–11.
5. Gasparetto, A.; Scalera, L. A brief history of industrial robotics in the 20th century. *Adv. Hist. Stud.* **2019**, *8*, 24–35. [CrossRef]
6. Collaborative Robot Applications. Available online: https://www.universal-robots.com/applications (accessed on 10 December 2022).
7. Knudsen, M.; Kaivo-Oja, J. Collaborative robots: Frontiers of current literature. *J. Intell. Syst. Theory Appl.* **2020**, *3*, 13–20. [CrossRef]
8. Bi, Z.M.; Luo, C.; Miao, Z.; Zhang, B.; Zhang, W.J.; Wang, L. Safety assurance mechanisms of collaborative robotic systems in manufacturing. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 102022. [CrossRef]
9. Fischer, S. Robotics—Market Data Analysis & Forecasts. Statista Technology Market Outlook. 2022. Available online: https://www.statista.com/study/116601/robotics-market-data-analysis-and-forecasts (accessed on 2 October 2022).
10. Roehl, C. Know Your Machine: Traditional Industrial Robots vs. Cobots. 2022. Available online: https://www.universal-robots.com/blog/know-your-machine-traditional-industrial-robots-vs-cobots (accessed on 12 January 2023).
11. Vojić, S. Applications of collaborative industrial robots. *Mach. Technol. Mater.* **2020**, *14*, 96–99.
12. Dmytriyev, Y.; Carnevale, M.; Giberti, H.; Todeschini, G. On cobot programming in industrial tasks: A test case. In Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Ankara, Turkey, 9–11 June 2022; IEEE: New York, NY, USA, 2022; pp. 1–9.
13. El Zaatari, S.; Marei, M.; Li, W.; Usman, Z. Cobot programming for collaborative industrial tasks: An overview. *Robot. Auton. Syst.* **2019**, *116*, 162–180. [CrossRef]
14. Schou, C.; Andersen, R.S.; Chrysostomou, D.; Bøgh, S.; Madsen, O. Skill-based instruction of collaborative robots in industrial settings. *Robot. Comput.-Integr. Manuf.* **2018**, *53*, 72–80. [CrossRef]
15. Schoen, A.; White, N.; Henrichs, C.; Siebert-Evenstone, A.; Shaffer, D.; Mutlu, B. CoFrame: A System for Training Novice Cobot Programmers. In Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction, Sapporo, Japan, 7–10 March 2022; IEEE: New York, NY, USA, 2022; pp. 185–194.
16. Kildal, J.; Tellaeche, A.; Fernández, I.; Maurtua, I. Potential users' key concerns and expectations for the adoption of cobots. *Procedia CIRP* **2018**, *72*, 21–26. [CrossRef]
17. Aaltonen, I.; Salmi, T. Experiences and expectations of collaborative robots in industry and academia: Barriers and development needs. *Procedia Manuf.* **2019**, *38*, 1151–1158. [CrossRef]
18. Kopp, T.; Baumgartner, M.; Kinkel, S. Success factors for introducing industrial human-robot interaction in practice: An empirically driven framework. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 685–704. [CrossRef]
19. Giannopoulou, G.; Borrelli, E.M.; McMaster, F. "Programming-It's not for Normal People": A Qualitative Study on User-Empowering Interfaces for Programming Collaborative Robots. In Proceedings of the 2021 30th IEEE International Conference

on Robot & Human Interactive Communication, Vancouver, BC, Canada, 8–12 August 2021; IEEE: New York, NY, USA, 2021; pp. 37–44.
20. Bogue, R. Europe continues to lead the way in the collaborative robot business. *Ind. Robot. Int. J.* **2016**, *43*, 6–11. [CrossRef]
21. Schumacher, S.; Hall, R.; Waldman-Brown, A.; Sanneman, L. Technology Adoption of Collaborative Robots for Welding in Small and Medium-sized Enterprises: A Case Study Analysis. In Proceedings of the Conference on Production Systems and Logistics, Vancouver, BC, Canada, 17–20 May 2022; Publish-Ing: Hannover, Germany, 2022; pp. 462–471.
22. Fantini, P.; Pinzone, M.; Sella, F.; Taisch, M. Collaborative robots and new product introduction: Capturing and transferring human expert knowledge to the operators. In *Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future, Proceedings of the AHFE 2017 International Conference on Human Aspects of Advanced Manufacturing, Los Angeles, CA, USA, 17–21 July 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; Volume 8, pp. 259–268.
23. Pieskä, S.; Kaarela, J.; Mäkelä, J. Simulation and programming experiences of collaborative robots for small-scale manufacturing. In Proceedings of the 2018 2nd International Symposium on Small-scale Intelligent Manufacturing Systems, Cavan, Ireland, 16–18 April 2018; IEEE: New York, NY, USA, 2018; pp. 1–4.
24. Ionescu, T.B.; Schlund, S. A participatory programming model for democratizing cobot technology in public and industrial Fablabs. *Procedia CIRP* **2019**, *81*, 93–98. [CrossRef]
25. Schmidbauer, C.; Schlund, S.; Ionescu, T.B.; Hader, B. Adaptive task sharing in human-robot interaction in assembly. In Proceedings of the 2020 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 14–17 December 2020; IEEE: New York, USA, 2020; pp. 546–550.
26. Owen-Hill, A. What Are the Different Programming Methods for Robots? Available online: https://blog.robotiq.com/what-are-the-different-programming-methods-for-robots (accessed on 19 October 2022).
27. ABB Robotics Operating Manual Introduction to RAPID. Available online: http://rovart.cimr.pub.ro/docs/OpIntroRAPID.pdf (accessed on 4 October 2022).
28. Universal Robots. The URScript Programming Language. Available online: https://s3-eu-west-1.amazonaws.com/ur-support-site/18383/scriptmanual_en_1.3.pdf (accessed on 4 October 2022).
29. Gusan, V.; Țîțu, M.A.; Oprean, C. Industrial robots versus collaborative robots—The place and role in non-conventional technologies. *Acta Tech. Napoc.-Ser. Appl. Math. Mech. Eng.* **2022**, *65*, 101–110.
30. Michaelis, J.E.; Siebert-Evenstone, A.; Shaffer, D.W.; Mutlu, B. Collaborative or simply uncaged? understanding human-cobot interactions in automation. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–12.
31. Tilley, J. *Automation, Robotics, and the Factory of the Future*; McKinsey & Company: Los Angeles, CA, USA, 2017.
32. Simoes, A.C.; Lucas Soares, A.; Barros, A.C. Drivers impacting cobots adoption in manufacturing context: A qualitative study. In *Advances in Manufacturing II: Volume 1— Solutions for Industry 4.0*; Springer: Cham, Switzerland, 2019; pp. 203–212.
33. Trattner, A.; Hvam, L.; Forza, C.; Herbert-Hansen, Z.N.L. Product complexity and operational performance: A systematic literature review. *CIRP J. Manuf. Sci. Technol.* **2019**, *25*, 69–83. [CrossRef]
34. Transeth, A.A.; Stepanov, A.; Linnerud, Å.S.; Ening, K.; Gjerstad, T. Competitive high variance, low volume manufacturing with robot manipulators. In Proceedings of the 2020 3rd International Symposium on Small-Scale Intelligent Manufacturing Systems, Gjovik, Norway, 10–12 June 2020; IEEE: New York, USA, 2020; pp. 1–7.
35. Kootbally, Z. Industrial robot capability models for agile manufacturing. *Ind. Robot. Int. J.* **2016**, *43*, 481–494. [CrossRef]
36. Harbers, M.; Peeters, M.M.; Neerincx, M.A. Perceived autonomy of robots: Effects of appearance and context. In Proceedings of the A World with Robots: International Conference on Robot Ethics, Lisbon, Portugal, 23–24 October 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 19–33.
37. Khamis, A.; Hussein, A.; Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. *Coop. Robot. Sens. Netw.* **2015**, *2015*, 31–51.
38. Ho, M.; Farid, A.; Majumdar, A. Towards a Framework for Comparing the Complexity of Robotic Tasks. In Proceedings of the Algorithmic Foundations of Robotics XV: Proceedings of the Fifteenth Workshop on the Algorithmic Foundations of Robotics. University of Maryland, College Park, MD, USA, 22–24 June 2022; Springer International Publishing: Cham, Switzerland, 2022; pp. 273–293.
39. Bengoa, P.; González-Ojeda, I.D.; Ibarguren, A.; Goenaga, B.; Martínez-De-Lahidalga, S.; Gkournelos, C.; Lotsaris, K.; Angelakis, P.; Makris, S.; Antolín-Urbaneja, J.C. Coordination of Two Robots for Manipulating Heavy and Large Payloads Collaboratively: SOFOCLES Project Case Use. In *Advances and Applications in Computer Science, Electronics, and Industrial Engineering, Proceedings of the Conference on Computer Science, Electronics and Industrial Engineering (CSEI 2021), Ambato, Ecuador, 26 May 2022*; Springer International Publishing: Cham, Switzerland, 2022; pp. 255–271.
40. Chromjakova, F.; Trentesaux, D.; Kwarteng, M.A. Human and cobot cooperation ethics: The process management concept of the production workplace. *J. Compet.* **2021**, *13*, 21–38. [CrossRef]
41. Alitappeh, R.J.; Jeddisaravi, K. Multi-robot exploration in task allocation problem. *Appl. Intell.* **2022**, *52*, 2189–2211. [CrossRef]
42. Seenu, N.; Kuppan Chetty, R.M.; Ramya, M.M.; Janardhanan, M.N. Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems. *Ind. Robot. Int. J. Robot. Res. Appl.* **2020**, *47*, 929–942.

43. Schneider, E.; Sklar, E.I.; Parsons, S. Mechanism selection for multi-robot task allocation. In Proceedings of the Towards Autonomous Robotic Systems: 18th Annual Conference, TAROS 2017, Guildford, UK, 19–21 July 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 421–435.
44. Murugappan, E.; Subramanian, N.; Rahman, S.; Goh, M.; Chan, H.K. Performance analysis of clustering methods for balanced multi-robot task allocations. *Int. J. Prod. Res.* **2022**, *60*, 4576–4591. [CrossRef]
45. Semwal, T.; Jha, S.S.; Nair, S.B. On ordering multi-robot task executions within a cyber physical system. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **2017**, *12*, 1–27. [CrossRef]
46. Sathyan, A.; Ma, O. Collaborative control of multiple robots using genetic fuzzy systems. *Robotica* **2019**, *37*, 1922–1936. [CrossRef]
47. Johnson, L.B.; Choi, H.L.; How, J.P. The role of information assumptions in decentralized task allocation: A tutorial. *IEEE Control. Syst. Mag.* **2016**, *36*, 45–58.
48. Cutting Edge Industrial Collaborative Robots, Built to Do More. Available online: https://www.universal-robots.com/e-series/ (accessed on 15 December 2022).
49. Karami, H.; Darvish, K.; Mastrogiovanni, F. A task allocation approach for human-robot collaboration in product defects inspection scenarios. In Proceedings of the 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 31 August–4 September 2020; IEEE: New York, NY, USA, 2020; pp. 1127–1134.
50. Lee, M.L.; Behdad, S.; Liang, X.; Zheng, M. Task allocation and planning for product disassembly with human–robot collaboration. *Robot. Comput.-Integr. Manuf.* **2022**, *76*, 102306. [CrossRef]
51. Roncone, A.; Mangin, O.; Scassellati, B. Transparent role assignment and task allocation in human robot collaboration. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; IEEE: New York, NY, USA, 2017; pp. 1014–1021.
52. Müller, R.; Vette, M.; Geenen, A. Skill-based dynamic task allocation in human-robot-cooperation with the example of welding application. *Procedia Manuf.* **2017**, *11*, 13–21. [CrossRef]
53. Liau, Y.Y.; Ryu, K. Task allocation in human-robot collaboration (HRC) based on task characteristics and agent capability for mold assembly. *Procedia Manuf.* **2020**, *51*, 179–186. [CrossRef]
54. Liau, Y.Y.; Ryu, K. Genetic algorithm-based task allocation in multiple modes of human–robot collaboration systems with two cobots. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 7291–7309. [CrossRef]
55. Ranz, F.; Hummel, V.; Sihn, W. Capability-based task allocation in human-robot collaboration. *Procedia Manuf.* **2017**, *9*, 182–189. [CrossRef]
56. Malik, A.A.; Bilberg, A. Complexity-based task allocation in human-robot collaborative assembly. *Ind. Robot. Int. J. Robot. Res. Appl.* **2019**, *9*, 182–189. [CrossRef]
57. Johannsmeier, L.; Haddadin, S. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robot. Autom. Lett.* **2016**, *2*, 41–48. [CrossRef]
58. Evangelou, G.; Dimitropoulos, N.; Michalos, G.; Makris, S. An approach for task and action planning in human–robot collaborative cells using AI. *Procedia Cirp* **2021**, *97*, 476–481. [CrossRef]
59. Yu, T.; Huang, J.; Chang, Q. Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning. *J. Manuf. Syst.* **2021**, *1*, 487–499. [CrossRef]
60. Make the Simple Tasks Easy and the Complex Tasks Possible. Available online: https://www.universal-robots.com/products/polyscope (accessed on 15 December 2022).
61. Offline Simulator—E-Series—URSim for non Linux 5.12.5. Available online: https://www.universal-robots.com/download/software-e-series/simulator-non-linux/offline-simulator-e-series-ur-sim-for-non-linux-5125/ (accessed on 30 November 2022).
62. Wizard Easy Programming. Available online: https://new.abb.com/products/robotics/application-software/wizard (accessed on 15 December 2022).
63. Technical Reference Manual. RAPID Instructions, Functions and Data Types. Available online: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf (accessed on 15 December 2022).
64. RobotStudio. Available online: https://new.abb.com/products/robotics/robotstudio/robotstudio-desktop (accessed on 15 December 2022).
65. 5 Key Reasons Why the Fanuc CRX Collaborative Robot Is Right for Everyone. Available online: https://crx.fanucamerica.com/why-cobots-collaborative-robots/ (accessed on 15 December 2022).
66. Programming a Robot Is Child's Play. Available online: https://www.festo.com/us/en/e/journal/programming-a-robot-as-child-s-play-id_28594/ (accessed on 16 December 2022).
67. iiQKA: Robots for the People. Available online: https://www.kuka.com/en-au/future-production/iiqka-robots-for-the-people (accessed on 16 December 2022).
68. User-Friendly Touchscreen Teach Pendant for Novice Robot Users. Available online: https://www.motoman.com/en-us/products/programming/smart-pendant (accessed on 16 December 2022).
69. HC Series Cobots—Your New Work Assistants. Available online: https://www.yaskawa.eu.com/products/robots/cobot (accessed on 16 December 2022).
70. Motoman NX100 Controller. Inform II User's Manual. Available online: http://www.wtech.com.tw/public/download/manual/yaskawa/NX100/YASKAWA%20NX100%20INFORM%20II%20USER%20Manual.pdf (accessed on 3 January 2023).

71. Automation Control Environment (ACE) Version 4 User Manual. Available online: https://www.edata.omron.com.au/eData/Robotics/I633-E-04.pdf (accessed on 10 December 2022).
72. RoboDK Basic Guide. Available online: https://robodk.com/doc/en/Basic-Guide.html (accessed on 4 January 2023).
73. Why Is It Worth Using ... ArtiMinds Robot Programming Software Solutions? Available online: https://www.artiminds.com/robot-programming-software/ (accessed on 18 December 2022).
74. Wandelbots. Everyone Can Work with Robots. Available online: https://wandelbots.com/en/ (accessed on 5 January 2023).
75. Pickit. Guide your Robot. Available online: https://www.pickit3d.com/en (accessed on 5 January 2023).
76. Robotmaster Offline Programming Software for Robots. Available online: https://www.robotmaster.com/en (accessed on 6 January 2023).
77. Universal Robots. Simplify Robot Programming with G-Code. Available online: https://www.universal-robots.com/blog/simplify-robot-programming-with-g-code/ (accessed on 5 January 2023).
78. ROS—Robot Operating System. Available online: https://www.ros.org/ (accessed on 7 January 2023).
79. Plant Automation Technology. Different Types of Robot Programming Languages. Available online: https://www.plantautomation-technology.com/articles/different-types-of-robot-programming-languages (accessed on 12 December 2022).
80. U.S. Social Security Administration. Code Of Federal Regulations. 404.1568. Skills Requirements. Available online: https://www.ssa.gov/OP_Home/cfr20/404/404-1568.htm (accessed on 2 February 2023).
81. Schou, C.; Madsen, O. A plug and produce framework for industrial collaborative robots. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417717472. [CrossRef]
82. Heimann, O.; Guhl, J. Industrial robot programming methods: A scoping review. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, Vienna, Austria, 8–11 September 2020; IEEE: New York, NY, USA, 2020; Volume 1, pp. 696–703.
83. Weintrop, D.; Afzal, A.; Salac, J.; Francis, P.; Li, B.; Shepherd, D.C.; Franklin, D. Evaluating CoBlox: A comparative study of robotics programming environments for adult novices. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; pp. 1–12.
84. Codejig. Block Coding. Available online: https://www.codejig.com/en/block-based-coding (accessed on 10 January 2023).
85. Fogli, D.; Gargioni, L.; Guida, G.; Tampalini, F. A Hybrid Approach to User-Oriented Programming of Collaborative Robots. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102234. [CrossRef]
86. GitHub. Your AI Pair Programmer GitHub Copilot. Available online: https://github.com/features/copilot/ (accessed on 17 February 2023).
87. Borboni, A.; Reddy, K.V.; Elamvazuthi, I.; AL-Quraishi, M.S.; Natarajan, E.; Azhar Ali, S.S. The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works. *Machines* **2023**, *11*, 111. [CrossRef]
88. Hiltner, J. Self-learning Intelligence for Object Recognition. *Quality*, 1 July 2019.
89. De Marchi, M.; Gualtieri, L.; Rojas, R.A.; Rauch, E.; Cividini, F. Integration of an Artificial Intelligence Based 3D Perception Device into a Human-Robot Collaborative Workstation for Learning Factories. In Proceedings of the 11th Conference on Learning Factories (CLF), Graz, Austria, 31 June–2 July 2021.
90. Bai, Y.; Lindqvist, B.; Karlsson, S.; Kanellakis, C.; Nikolakopoulos, G. Multi-Robot Task Allocation Framework with Integrated Risk-Aware 3D Path Planning. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 28 June–1 July 2022; IEEE: New York, NY, USA, 2022; pp. 481–486.
91. Kaczmarek, W.; Panasiuk, J.; Borys, S.; Banach, P. Industrial robot control by means of gestures and voice commands in off-line and on-line mode. *Sensors* **2020**, *20*, 6358. [CrossRef]
92. Mendes, N.; Safeea, M.; Neto, P. Flexible programming and orchestration of collaborative robotic manufacturing systems. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics, Porto, Portugal, 18–20 July 2018; IEEE: New York, NY, USA, 2018; pp. 913–918.
93. Pérez, L.; Diez, E.; Usamentiaga, R.; García, D.F. Industrial robot control and operator training using virtual reality interfaces. *Comput. Ind.* **2019**, *109*, 114–120. [CrossRef]
94. Psarakis, L.; Nathanael, D.; Marmaras, N. Fostering short-term human anticipatory behavior in human-robot collaboration. *Int. J. Ind. Ergon.* **2022**, *87*, 103241. [CrossRef]
95. De Pace, F.; Manuri, F.; Sanna, A.; Fornaro, C. A systematic review of Augmented Reality interfaces for collaborative industrial robots. *Comput. Ind. Eng.* **2020**, *149*, 106806. [CrossRef]
96. Ong, S.K.; Yew, A.W.W.; Thanigaivel, N.K.; Nee, A.Y. Augmented reality-assisted robot programming system for industrial applications. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101820. [CrossRef]
97. Introducing Wizard Easy Programming. Available online: https://youtu.be/2l-IKmdcJsM (accessed on 18 December 2022).
98. Omron Adept Robot Programming: It's That Easy! Available online: https://www.youtube.com/watch?v=9Y-3i0I07Bc (accessed on 18 December 2022).
99. Robotiq. Skills to Integrate a Robot. Available online: https://blog.robotiq.com/skills-to-integrate-a-robot?_ga=2.195912812.1784018681.1676172069-1475687650.1676171346 (accessed on 30 January 2023).
100. Doyle-Kent, M.; Kopacek, P. Adoption of collaborative robotics in industry 5.0. An Irish industry case study. *IFAC-Pap.* **2021**, *54*, 413–418. [CrossRef]

101. Gjeldum, N.; Aljinovic, A.; Crnjac Zizic, M.; Mladineo, M. Collaborative robot task allocation on an assembly line using the decision support system. *Int. J. Comput. Integr. Manuf.* **2022**, *35*, 510–526. [CrossRef]
102. Universal Robots. The UR5e. Available online: https://www.universal-robots.com/products/ur5-robot/ (accessed on 10 October 2022).
103. Park, J.; Jung, W. A study on the development of a task complexity measure for emergency operating procedures of nuclear power plants. *Reliab. Eng. Syst. Saf.* **2007**, *92*, 1102–1116. [CrossRef]
104. Park, J.; Jang, I. Can we determine the complexity level of a proceduralized task? In Proceedings of the Transactions of the Korean Nuclear Society Virtual Spring Meeting, Jeju, Korea, 13–14 May 2021.
105. Kirschgens, L.A.; Ugarte, I.Z.; Uriarte, E.G.; Rosas, A.M.; Vilches, V.M. Robot hazards: From safety to security. *arXiv* **2018**, arXiv:1806.06681.

applied sciences

MDPI

Article

Switch-Off Policies in Job Shop Controlled by Workload Control Concept

Paolo Renna

School of Engineering, University of Basilicata, 85100 Potenza, Italy; paolo.renna@unibas.it; Tel.: +39-0971205143

Abstract: The reduction in emissions and the increase in energy costs push companies to identify solutions to reduce energy consumption in production systems. One of the approaches proposed in the literature is the shutdown of machines to reduce energy consumption in the idle state. This solution does not affect production processes and can be applied in various manufacturing fields. This paper proposes switch-off policies in manufacturing systems under a workload control system. The shutdown policies developed consider the number of items in the queue and the calculation derived from the workload control mechanism. Simulation models have been developed to test the proposed policies using the case always on as a benchmark, considering different levels of absorbed power in the inactivity and warm-up states and different warm-up times. The results highlight how the switch policies that include the workload evaluation drastically reduce the number of on/off activities, assuring lower energy consumption.

Keywords: sustainable manufacturing; switch-off; workload control; simulation

1. Introduction

Environmental sustainability and energy costs are crucial topics in manufacturing systems. The importance of identifying new solutions to improve the energy efficiency of manufacturing systems is essential to reduce the emission of greenhouse gases and the energy bill. Energy savings is the main way to meet the climate change targets set by countries around the world [1]. About 24% and 5% of global greenhouse gas emissions are related, respectively, to industrial energy consumption and industrial processes [2]. Then, the success of the reduction in energy consumption and increment in renewable energy sources depends strongly on industrial energy efficiency [3]. Moreover, the installed power of renewable energy sources grew from 2011 to 2020 continuously [4], with the main contribution by solar and wind energy, about 91%. This expansion is also due to the reduction in installation costs of renewable sources [5]. Among the methodologies proposed for the reduction of energy consumption, the switch-off approach [6] is a promising strategy. This approach does not change the manufacturing processes, and no new technology or expensive equipment is necessary. The switch-off policy works like the start and stop of the cars to reduce the energy consumed in the idle state. The fields of industrial applications can be different, such as CNC machining operation, welding, plastic deformation, and many other manufacturing processes.

The switch-off policies are mainly proposed for flow lines [7,8], but few works have studied these policies in job-shop manufacturing systems [9]. The job-shop systems are characterized by a variable routing of the parts for which it is more complex to introduce switch-off policies than in flow line systems. The introduction of a switch-off policy in job-shop systems needs to use information about the system as the production planning model. A production planning control used for job shops that work in make-to-order is the WorkLoad Control (WLC) approach [10,11].

The switch-off policies proposed in the literature evaluate the upstream, downstream, or both buffers for production lines. The research proposes the introduction of switch-off policies in job-shop systems using the information of the WorkLoad Control method.

The introduction of the WLC method allows the evaluation of the direct workload in the queues of the machines with the indirect workload (WLC) to improve the switch-off/on decisions. It aims to pursue a trade-off between energy reduction and manufacturing system performance. Simulation models are used to test the proposed switch-off policies considering the manufacturing performance, energy reduction, and the number of on/off activities. The number of switches on/off activities is not studied in the literature but can be relevant for the influence on the reliability of the machines.

This paper is organized as follows. Section 2 discusses the literature review about the switch-off policies in manufacturing systems. Section 3 describes the manufacturing system context under the workload control policy. Section 4 explains the switch-off policies based on workload computation. The simulation experiments and the numerical results are discussed in Section 5. Section 6 provides the conclusions and future research path.

2. Literature Review

Numerous works have been proposed in the literature on workload control and the switch-off in production lines. The following discussion of the literature concerns the proposed switch-off policies and applications in job-shop systems.

The switch-off policies have mainly been proposed in flow line systems [7,12]. These approaches work on three decision evaluations: supervising the upstream buffer level; supervising the downstream buffer level; and supervising the upstream and downstream buffer levels together. These approaches reduce the energy consumed in the idle state of the machines. The effectiveness of these models has also been tested in pull control systems of flow lines [8].

The development of mathematical models can support the introduction of switch-off policies, but this approach can increase computational complexity with the problem of applying these models in real industrial applications [13].

To reduce the computational complexity, a fuzzy controller that supports each machine that collects the real-time data to switch off/on the machines was proposed in the flow line [14,15].

Ref. [16] studied the switch-off policy introduction in the design model of the flow line. This design model introduces a processing time distance for each couple of stations to facilitate the switch-off of the machines. The numerical results show how this approach leads to a significant reduction in energy consumption, limiting production loss.

Few works have been developed in the field of job shop systems. The authors of [17] include the switch-off strategy using a mixed-integer linear programming model to reduce energy consumption in a flexible job-shop system. The model proposed cannot identify energy-efficient production schedules for real industrial applications.

Ref. [18] studied the scheduling problem for flexible job shops considering the switch-off and speed processing time to save energy. A genetic algorithm is developed to solve the mathematical problem to optimize the makespan, the energy consumption, and the number of turning-on/off machines simultaneously.

Ref. [19] proposed a mixed-integer programming mode with a genetic algorithm to optimize the makespan and reduce energy consumption. The numerical results show a potential energy consumption reduction, but the genetic algorithm can lead to an increase in the computational complexity for industrial cases.

Ref. [9] proposed a model that combines the direct and indirect workload of the stations of a job shop. The numerical results highlight that it is possible to obtain a compromise between energy reduction consumption and production loss.

Ref. [20] proposed a novel mathematical formulation that includes switching off the machines to reduce energy consumption in flexible job shop scheduling problems. They introduced a decomposition approach to allow the application of the proposed model for large-scale problems.

Ref. [21] designed a fuzzy controller to switch off the machine while considering the upstream buffer level and the required production rate. The simulation experiments on a machine highlighted that a large amount of energy could be saved without affecting the throughput significantly. This approach could be extended to job shop systems with more than one machine.

Ref. [22] discussed the resilience of integrated energy systems that can be affected by the shutdown of the machines of a complete manufacturing system. The authors [23,24] highlighted how the Internet of Things (IoT) technologies and cloud computing improve the monitoring and operations of energy management systems.

The literature review analysis highlights the following limits:

- Few works have introduced switch-off policies in job-shop manufacturing systems. The majority of papers developed mathematical models that can be difficult to apply in real industrial applications.
- The introduction of switch-off policies in manufacturing systems controlled by a workload mechanism has not been studied in the literature.

In response, this paper studied the introduction of several switch-off policies in a job shop controlled by a workload mechanism by first asking (RQ1): what is the impact of the switch-off policies in a job-shop system controlled by the workload mechanism?

The switch-off policies can include a different mechanism to turn off/on the machines of the manufacturing system, and then our second research question asks (RQ2): what is the impact on the main performance measures of the combinations of the turn-off/on mechanism also based on the power consumed in the states of the machines?

3. Research Context

The proposed switch-off policies in a job shop controlled by workload mechanism were evaluated using the same model introduced in previous studies [25,26] and investigated in many works afterward. The main characteristics of the production system are briefly described below. The job shop consists of six work centers, and each work center includes one machine. The jobs enter the system following a random routing sequence without any preferred routing; then, a statistical processing time for each machine (as described in the simulation experiments section) and a due date are assigned to each job. According to previous works [27,28], all jobs are accepted, and raw materials are always available. Moreover, the main assumptions of the manufacturing system are the following: operations cannot be pre-empted; each machine can process only one task at once; the queues are managed by the Earliest Due Date (EDD) policy to improve the lateness performance; the machining time includes the material handling time; and the handling resources are always available. The notation used is described in the following:

Notation		Definition
Indices	M	The number of work centers/machines that compose the manufacturing system
	m	The index of the machines $m = 1, \ldots, M$
	i	The index of the jobs
Parameters	PT_{im}	The processing time of the job i in the machine m
	a_{im}	A binary value equal to 1 if the job i must visit the machine m; 0 otherwise
	DD_i	The due date assigned to the job i
	Seq_{im}	The ordered sequence of the machine m for the job i
	WLnorm	The norm of the workload control mechanism
Computation	WLm	Workload of the machine m

The workload control mechanism applied is a classical approach proposed in the literature [27] following a continuous order release and fixed workload norm according to the following steps. The jobs enter the pre-shop queue of the manufacturing system that is managed according to the Earliest Due Date rule. Then, starting from the first job in the pre-shop queue, the potential workload of each machine added by the job is computed. The potential workload is computed using the corrected aggregate load method [26], as shown in Expression (1):

$$WL_m = WL_m + \sum\nolimits_{m=1}^{M} \frac{PT_{im} * a_{im}}{Seq_{im}} \quad (1)$$

The corrected workload (Expression (1)) converts the load contribution considering that the processing time of an operation is divided by the position of the corresponding machine in the routing of the job. The potential workload computed supports the decision on the release of the job in the manufacturing system. The job can enter the manufacturing system if the workload of each machine is lower than the workload norm WLnorm. If the job is released, the workload computed as shown in Expression (1) is updated for each work center.

When job i leaves a machine, the workload of the machine is updated, as shown in Expression (2):

$$WL_m = WL_m - \frac{PT_{im}}{Seq_{im}} \quad (2)$$

The workload is updated considering the position of the machine in the routing of the job to keep correct the input/output of the workload computation.

4. Switch off Policies

A shutdown policy is characterized by two decisions: what is the condition for shutting down and the condition for turning on the machine? Figure 1 shows the activities to implement a switch-off policy; when a machine loads a part, the state runs until the end of the machining time. So, the machine is "on service" and can either load another part or shut down depending on the policy being enforced. If the machine shuts down, when the policy decides to turn it on, it is considered a warm-up period to transition to the "on service" state. The shutdown policy must consider when to shut down when the machine is in the "in service" state or when to power back on when the machine is in the out-of-service state.

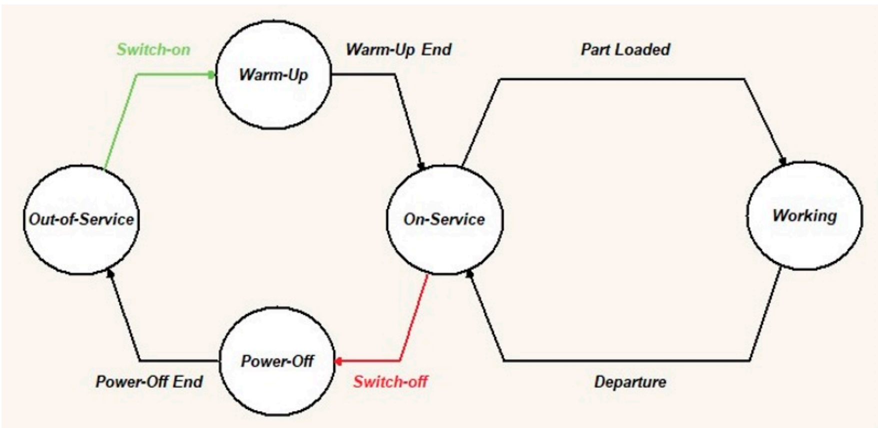


Figure 1. Switch-off control.

The approaches proposed in the literature often evaluate the level of the upstream and downstream buffer. In the case of a job shop system with the dynamic routing of the jobs, the upstream buffer can be evaluated, while the downstream buffer cannot be used because this buffer is not related exclusively to a specific machine as in the flow lines. Then, the information that can support the switch-off policies for each machine is the upstream buffer level and the workload computation derived from the workload control mechanism. The combination of this information is proposed for computing a modified workload that supports the switch-off policy. The modified workload is calculated as the combination of the workload and the items in the queue of the workstations. The modified workload is computed as shown in Expression (3); this expression concerns the weighted sum of two normalized values. The first is the workload of the machine m related to the workload norm, and the second is the number of parts in the queue related to the work in the process of the manufacturing system. The workload norm is the maximum value possible for the workload of the machines, and the WIP is the maximum number of parts in the manufacturing system.

$$WLmod_m = \alpha * \frac{WL_m}{WLnorm} + \beta * \frac{Queue_m}{WIP} \tag{3}$$

where WIP is the Work In Process of the jobs released in the manufacturing system and

$$\alpha + \beta = 1 \tag{4}$$

The modified workload, as shown in Equation (3), assumes values between 0 and 1. Combining the upstream buffer level, workload, and modified workload can obtain five switch-off policies, as shown in Table 1.

Table 1. Switch-off policies investigated.

Switch-Off Policy	Off Condition Machine m	On Condition Machine m
Policy 1	$Queue_m = 0$	$Queue_m = 1$
Policy 2	$WL_m < Threshold_1$	$Queue_m = 1$
Policy 3	$WLmod_m < Threshold_2$	$Queue_m = 1$
Policy 4	$WLmod_m < Threshold_3$	$WLmod_m > Threshold_4$
Policy 5	$WL_m < Threshold_5$	$WL_m > Threshold_6$

Policy 1 is used as the benchmark because it turns off the machines when the queue is empty and turns on as soon as an item arrives in the queue, and this is the upstream policy widely used in the literature [12].

Policy 2 uses the workload computed for the workload control to switch off the machine while the machine turns on when an item arrives in the queue. Then, the first threshold ($threshold_1$) should be defined. Policy 3 differs from policy 2 in terms of the switch-on condition that considers the workload modified, as computed in Equation (3) with another threshold ($threshold_2$) used to define.

The modified workload is used to switch on and off the machines for policy 4 with the relative thresholds ($Threshold_3$ and $Threshold_4$). Finally, policy 5 considers the workload computation to switch off/on with another two thresholds ($Threshold_5$ and $Threshold_6$).

5. Simulation Environment

The performance of the proposed switch-off policies is compared with the always-on model. The simulation model has the same characteristics as previous work proposed in the literature [27,28]; Table 2 reports the model characteristics for the simulations conducted.

Table 2. Model characteristics.

Number of machines	6, including 1 bottleneck
Inter-arrival	EXPO (0.642)
Number of operations	Discrete Uniform [1, 6]
Due date	(total processing time) $\times$ Uniform [5, 10]
Processing times, no bottlenecks	2-Erlang with mean 1
Processing times bottlenecks	2-Erlang with mean 1.15 (utilization about 90%)

The job shop consists of 6 work centers/machines with 1 bottleneck; the processing time of machines that have no bottlenecks follows a 2-Erlang distribution with a mean of 1 and a mean of 1.15 for the bottleneck to lead to an average utilization of 90%. The job's arrival follows an exponential distribution with a parameter of 0.642.

The routing of the jobs is random, without any preferential sequence, with the number of operations extracted by a discrete uniform between 1 and 6.

Due date is assigned to each job, considering the total processing time multiplied by a parameter extracted by a uniform distribution. Finally, the simulation length is 25,000 h.

The simulation model described above was developed using the software package SIMUL® (version 29.0).

Simul8 is a computer package for Discrete Event Simulation to simulate and model a wide variety of manufacturing systems, such as production lines, job shops, robotics cells, assembly systems, and complex product flows. The simulations conducted by SIMUL8 provide a series of statistics on the main performance measures of the manufacturing system tested.

The effects of the power in the three states (idle, stop, and warm-up) of the machines and the warm-up time are considered to evaluate the proposed switch-off policies. The power of the work state is fixed because the objective is to evaluate the relation between the power in the work state and the other states. In detail, 3 values for the warm-up time are considered, 0.2, 0.4, and 0.6, which correspond to 20%, 40%, and 60% of the mean processing time of the machines.

Table 3 shows the power cases evaluated for the states of work, idle, stop, and warm-up of the machines.

Table 3. Power sensitivity analysis.

Power	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
Work (Kw)	5	5	5	5	5	5	5	5	5
Idle (Kw)	3	4	5	3	4	5	3	4	5
Stop (kw)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Warm-up (Kw)	5	5	5	4	4	4	3	3	3

The experiments concern 6 models (benchmark and 5 switch-off policies), 9 cases of power values, and 3 warm-up times, with a total of 162 cases. The simulations are repeated for several values of the thresholds (see Table 1) to obtain results similar to policy 1 proposed in the literature.

For each class of experiment, a series of replicates were carried out capable of ensuring a confidence interval of 5% and 95% of the confidence level for each performance measure. Each combination of the experimental class features over 2000 replicas and approximately 6 h of computation time (4 GHz Intel Core i7 and 8 Gb RAM). The simulations evaluate the performance measures in the following areas: the performance of the manufacturing system: throughput (products/unit time), number of products delayed (products), total time of lateness (unit time), and the lateness for the product unit (unit time/product); the performance about the energy consumption includes the energy consumed in idle, working, stop and warm-up states.

6. Numerical Results

The simulations for the benchmark case are repeated for different values of the workload norm to obtain a better performance. The workload norm that leads to better performance is equal to 12 for all simulations conducted.

Table 4 reports the values of the thresholds, alfa, and beta that lead to performance similar to policy 1, which is the policy proposed in the literature.

Table 4. Parameters for the switch-off policies.

	Parameter	Value		Value
Policy 2	**Threshold$_1$**	1		
Policy 3	**Threshold$_2$**	0.05		
Policy 3	**alfa**	0.6	**beta**	0.4
Policy 4	**Threshold$_3$**	0.05	**Threshold$_4$**	1
Policy 5	**Threshold$_5$**	0.09	**Threshold$_6$**	1.1
	alfa	0.6	**beta**	0.4

Figure 2 shows the reduction in energy consumption compared to the case of always on for the five switch-off policies considering the nine cases of power consumption and three warm-up times.

Policy 1 always leads to better energy reduction for all cases tested. The increment in the warm-up times reduces the energy reduction consumption for all policies tested. This is due to the higher energy consumed during the warm-up period.

Cases 1, 4, and 7 lead to lower energy reduction; these cases are characterized by lower energy consumption in the idle state of the machines. The cases with higher energy consumption in the idle state (3, 6, and 9) improve energy reduction. Then, it is important to evaluate the characteristics of the machines to estimate the energy reduction in the switch-off policies.

Except for case 2, policy 4 leads to better energy reduction than the other policies that include workload computation.

Table 5 reports the ANOVA (α = 0.05) analysis conducted considering the idle, warm-up power, and warm-up time as the source of variance. The sources of variance are relevant for all policies tested except policy 5. The ANOVA analysis highlights how policy 5 is more robust in terms of the warm-up characteristics.

Figure 3 shows the impact of the main effects of energy reduction for the five policies studied. Policies 2, 3, 4, and 5 reduce the variability of the energy reduction compared to policy 1 based only on the queues.

Figure 4 shows the number of on/off activities that can affect the tear of the machines. Policy 2, 3, and 4 limit this value between 12,000 and 12,500. Policy 1 (based on the queues) increases the number of on/off activities, and this policy is more affected by the warm-up time.

Figure 5 shows the number of parts in delay compared to the model always being on. As per the previous measure, policies 2, 3, and 4 have better performance, while policy 1 is the worst, except when the warm-up time is lower.

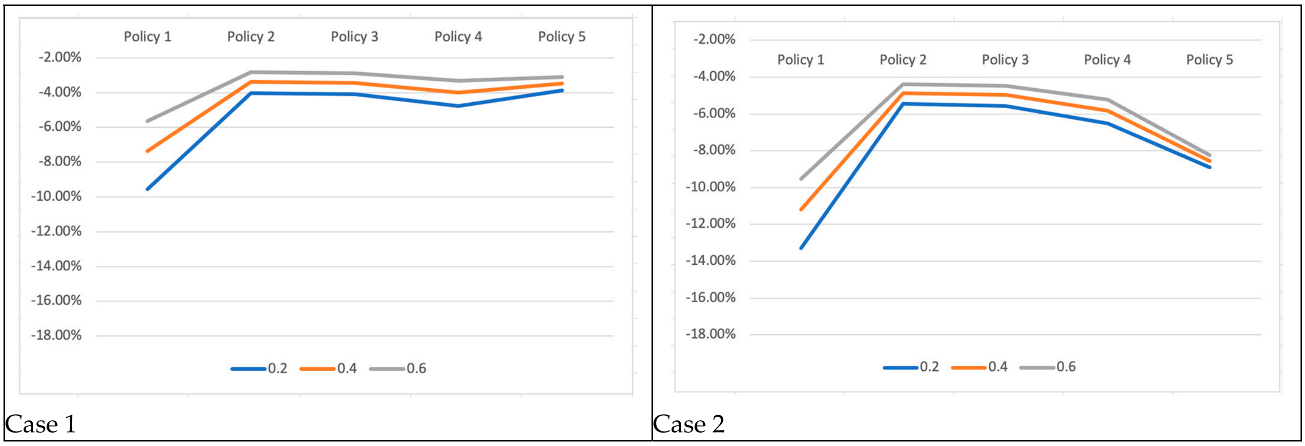


Figure 2. *Cont.*

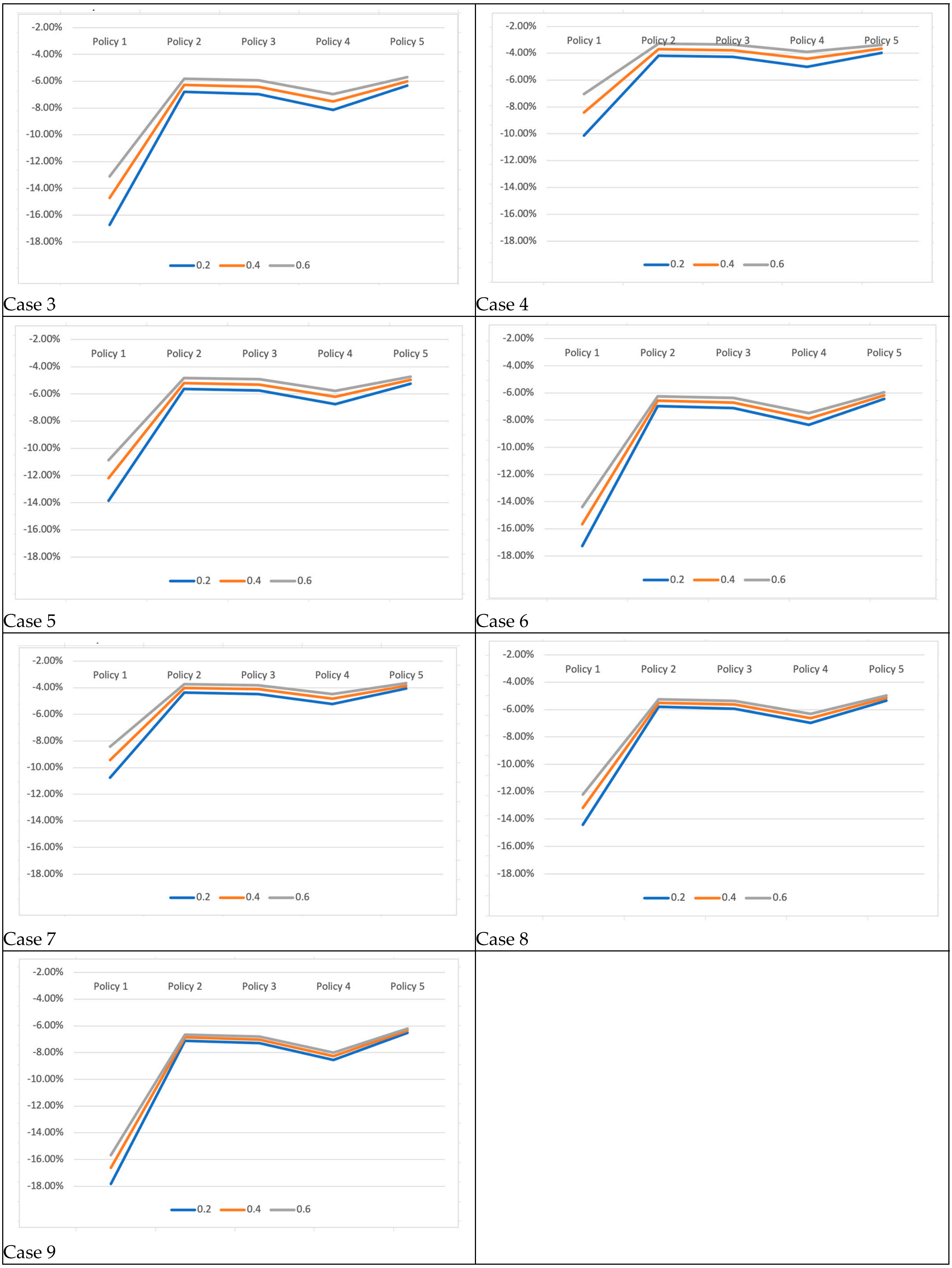


Figure 2. Energy consumption reduction.

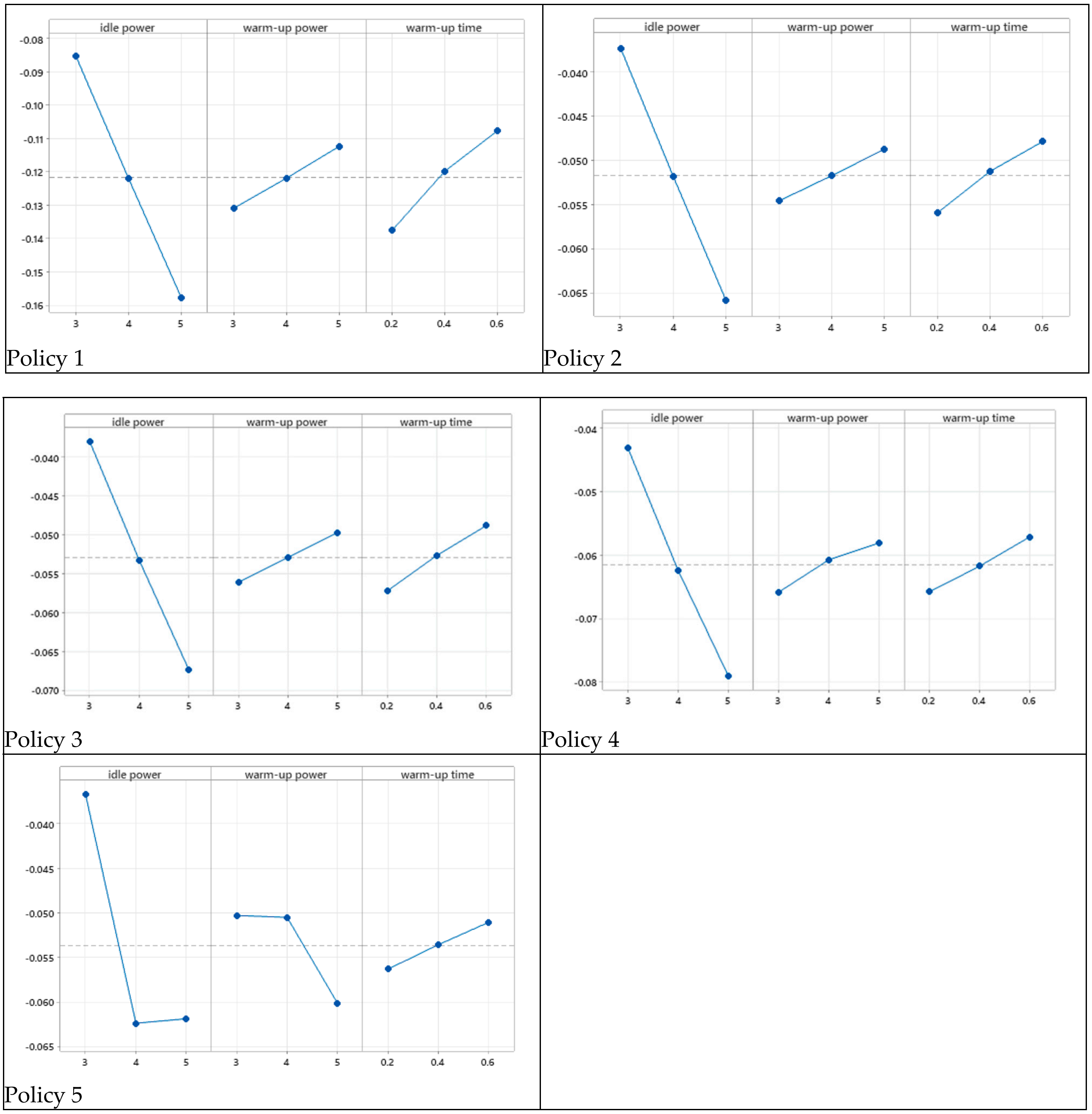


Figure 3. Main effects.

The same consideration of the part delayed can be confirmed by the total time delay, as shown in Figure 6.

Figure 7 shows the time delay for the unit of the part delayed. The improvement in this performance means that the increment in total time delay is not proportional to the increment in parts delayed. Then, the parts delayed increase with switch-off policies' increase with lower delay time accumulated. This can be important if the penalty is related to the time delay of the parts.

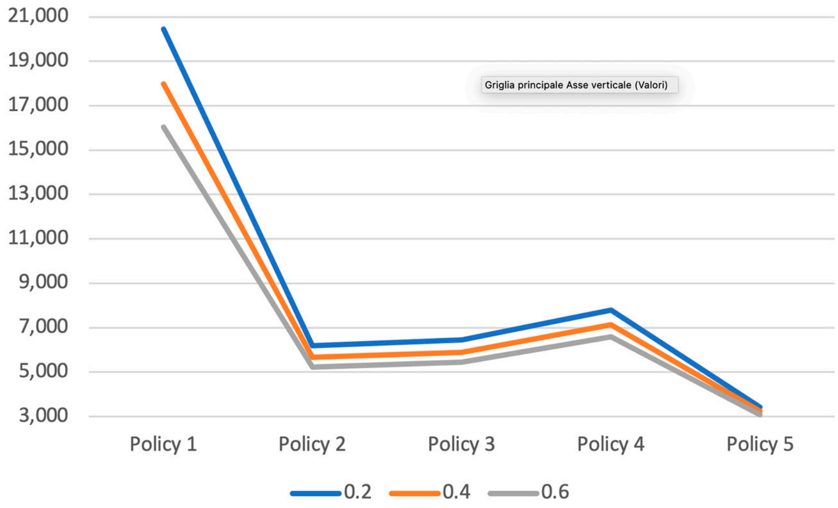


Figure 4. Number of on/off activities.

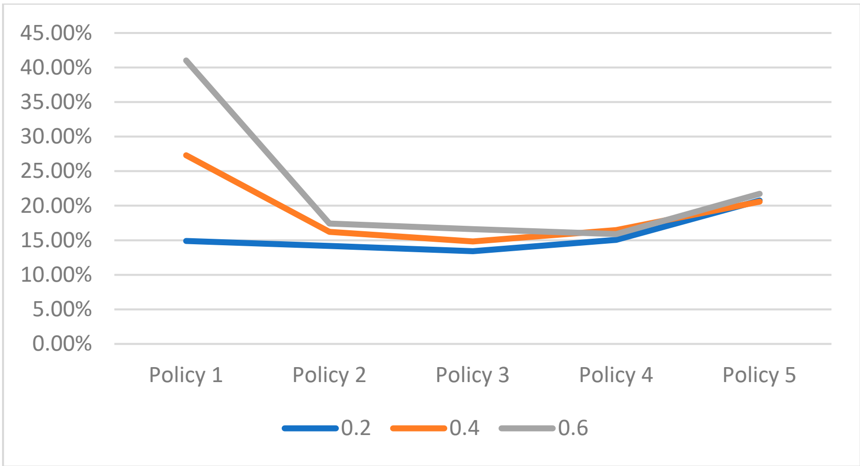


Figure 5. Parts delayed.

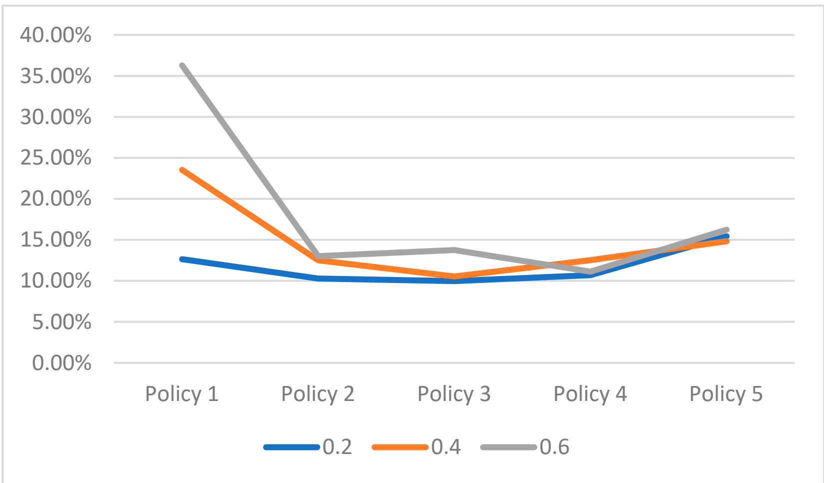


Figure 6. Total time delay.

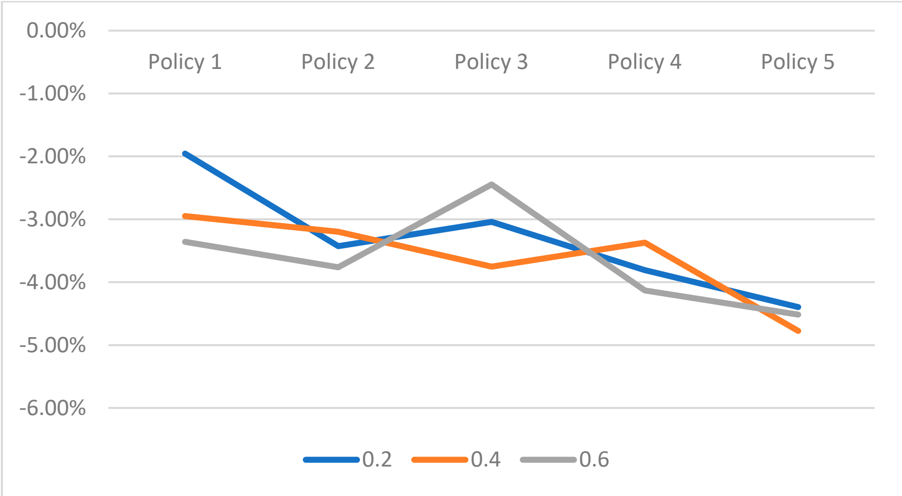


Figure 7. Time delay for unit of product.

Table 5. ANOVA analysis.

Source of Variance	Sum of Square	Degree of Freedom	Mean of Square	F-Ratio	*p*-Value
		Policy 1			
Idle power	0.023625	2	0.011813	911.52	0.000
Warm-up power	0.001617	2	0.000808	62.38	0.000
Warm-up time	0.004072	2	0.002036	157.12	0.000
Residual	0.000259	20	0.000013		
		Policy 2			
Idle power	0.003696	2	0.001848	1180.47	0.000
Warm-up power	0.000163	2	0.000082	52.07	0.000
Warm-up time	0.000299	2	0.000149	95.40	0.000
Residual	0.000031	20	0.000002		
		Policy 3			
Idle power	0.003892	2	0.001946	1542.30	0.000
Warm-up power	0.000184	2	0.000092	72.79	0.000
Warm-up time	0.000320	2	0.000160	126.80	0.000
Residual	0.000025	20	0.000001		
		Policy 4			
Idle power	0.005817	2	0.002909	505.34	0.000
Warm-up power	0.000280	2	0.000140	24.35	0.000
Warm-up time	0.000338	2	0.000169	29.33	0.000
Residual	0.000115	20	0.000006		
		Policy 5			
Idle power	0.003873	2	0.001937	20.23	0.000
Warm-up power	0.000573	2	0.000286	2.99	0.073
Warm-up time	0.000125	2	0.000062	0.65	0.532
Residual	0.001914	20	0.000096		

The throughput of the systems does not change with the switch-off policies.
From the analysis of the results, the following points can be summarized:

- Policy 1, proposed in the literature, leads to a greater reduction in energy consumption; the main limitations of this policy are greater variability in performance and an extremely high number of machine on/off activities that can reduce the reliability of the machines. Moreover, the production performance measures are worst in these cases.

- Among the policies proposed, policy 4 (based on the modified workload computation) leads to the best compromise of energy reduction, production performance measures, and number of machine-on/off activities.
- The ANOVA highlights how policy 5 is the more robust to the change in the parameters studied.
- The simulation model is a crucial method used to estimate the performance of a switch-off policy from several points of view.

7. Conclusions and Future Developmental Paths

The research proposed in this paper extends the switch-off method to reduce energy consumption in job-shop systems. A production control method used in the job shop system is workload control; in the literature, the effects of the switch-off policies in these systems were not studied.

Then, the research proposed in this paper introduces the switch-off method in manufacturing systems controlled by a workload control approach to reduce energy consumption. In response, our first research question asked: what is the impact of switch-off policies in a job-shop system controlled by the workload mechanism?

The simulation results have demonstrated how the switch-off policies can reduce the energy consumption of the manufacturing system by reducing the energy consumed in the idle state of the machines. The upstream policy, proposed in the literature, allows for drastically reducing energy consumption, but the number of turns on/off of the machines is very high, and the performance measures of the manufacturing system are the worst.

Then, the policies proposed that include the workload computation of the control mechanism allow us to obtain a better trade-off between energy consumption and manufacturing system performance. Moreover, the number of turns on/off of the machines is lower with the proposed switch-off policies.

The analysis of the different values of idle power and warm-up characteristics answers our second research question: what is the impact on the main performance measures of the combinations of the turn-off/on mechanism also based on the power consumed in the states of the machines?

The idle and warm-up power characteristics impact the manufacturing performance and energy consumption, as shown by the ANOVA analysis. The simulations show how the proposed policy based on workload computation is more robust against these parameters. Moreover, policy 5, based on workload control data, is not affected by the warm-up power and time.

At the managerial level, the simulation supports the decision maker in choosing the better switch-off policy for energy consumption reduction and the manufacturing performance target. The simulation helps the decision maker because it allows for estimating both the productivity and energy performance of the manufacturing systems. The potential industrial applications can involve production systems where CNC machines are used. Recent CNC machines are capable of switching into energy-saving modes or even shutting down completely. For example, [29] argued how in an aircraft small-parts supplier, there is an idle period of 16% of the machines, and this can reduce the energy consumption by about 13% with a switch-off policy. Therefore, the proposed method can support industrial cases with several CNC machines (such as cutting operations) or auxiliary tools such as air compressed for welding tasks. This is because both the CNC machines and air-compressed auxiliary tools can easily be turned off and on.

This research, following the works proposed in the literature, concerns a manufacturing system with dedicated machines; a future development path can investigate the impact of machine flexibility. A limitation of the proposed method is that the flexibility of the machines is not considered, and the method works with a determined routing of the jobs. Another limit is the processing time, which should have lower variability to aid the proposed model. A future research path can investigate the impact of workload control and a switch-off policy on the peak power constraint due to the energy provider.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. De la Cruz-Lovera, C.; Perea-Moreno, A.-J.; De la Cruz-Fernández, J.-L.; Alvarez-Bermejo, J.A.; Manzano-Agugliaro, F. Worldwide Research on Energy Efficiency and Sustainability in Public Buildings. *Sustainability* **2017**, *9*, 1294. [CrossRef]
2. Ritchie, H.; Roser, M. CO_2 and Greenhouse Gas Emissions. In Our World in Data. 2020. Available online: https://web.archive.org/web/20220719060016/https:/ourworldindata.org/co2-and-other-greenhouse-gas-emissions (accessed on 1 June 2022).
3. Gerarden, T.D.; Newell, R.G.; Stavins, R.N. Assessing the Energy-Efficiency Gap. *J. Econ. Lit.* **2017**, *55*, 1486–1525. [CrossRef]
4. IRENA. *Renewable Capacity Statistics 2021*; International Renewable Energy Agency: Abu Dhabi, United Arab Emirates, 2021; ISBN 978-92-9260-342-7.
5. IRENA. *Future of Solar Photovoltaic: Deployment, Investment, Technology, Grid Integration and Socio-Economic Aspects*; International Renewable Energy Agency: Abu Dhabi, United Arab Emirates, 2019; ISBN 978-92-9260-156-0.
6. Frigerio, N.; Matta, A. Energy-Efficient Control Strategies for Machine Tools with Stochastic Arrivals. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 50–61. [CrossRef]
7. Frigerio, N.; Matta, A. Analysis of production lines with switch-Off/On controlled machines. In *Eco-Factories of the Future. Sustainable Production, Life Cycle Engineering and Management*; Thiede, S., Herrmann, C., Eds.; Springer: Cham, Switzerland, 2019; pp. 147–172. [CrossRef]
8. Renna, P. Energy saving by switch-off policy in a pull-controlled production line. *Sustain. Prod. Consum.* **2018**, *16*, 25–32. [CrossRef]
9. Renna, P.; Materi, S. Switch off policies in job-shop manufacturing systems including workload evaluation. *Int. J. Manag. Sci. Eng. Manag.* **2021**, *16*, 254–263. [CrossRef]
10. Fernandes, N.O.; Carmo-Silva, S. Workload control under continuous order release. *Int. J. Prod. Econ.* **2011**, *131*, 257–262. [CrossRef]
11. Stevenson, M.; Hendry, L.C.; Kingsman, B.G. A Review of Production Planning and Control: The Applicability of Key Concepts to the Make-to-Order Industry. One World? One View of Operations Management? In Proceedings of the EurOMA-POMS Conference, Como, Italy, 16–18 June 2003; Available online: https://eprints.lancs.ac.uk/id/eprint/46212/ (accessed on 29 March 2023).
12. Frigerio, N.; Matta, A. Analysis on energy efficient switching of machine tool with stochastic arrivals and buffer information. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 238–246. [CrossRef]
13. Mashaei, M.; Lennartson, B. Energy reduction in a pallet- constrained flow shop through on–off control of idle machines. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 45–56. [CrossRef]
14. Wang, J.; Xue, J.; Duque, E.T.; Li, S.; Chang, Q. Fuzzy decision of machine switch on-off for energy efficient operation of manufacturing system. In Proceedings of the 13th IEEE Conference on Automation Science and Engineering, Xi'an, China, 20–23 August 2017.
15. Wang, J.; Fei, Z.; Chang, Q.; Li, S.; Fu, Y. Multi-state decision of unreliable machines for energy-efficient production considering work-in-process inventory. *Int. J. Adv. Manuf. Technol.* **2019**, *102*, 1009–1021. [CrossRef]
16. Renna, P.; Materi, S. Design model of flow lines to include switch-off policies reducing energy consumption. *Appl. Sci.* **2020**, *10*, 1475. [CrossRef]
17. Zhang, L.; Tang, Q.; Wu, Z.; Wang, F. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. *Energy* **2017**, *138*, 210–227. [CrossRef]
18. Wu, X.; Sun, Y. A green scheduling algorithm for flexible job shop with energy-saving measures. *J. Clean. Prod.* **2018**, *172*, 3249–3264. [CrossRef]
19. Liu, M.; An, L.; Zhang, J.; Chu, F.; Chu, C. Energy-oriented bi-objective optimisation for a multi-module reconfigurable manufacturing system. *Int. J. Prod. Res.* **2018**, *57*, 5974–5995. [CrossRef]
20. Rakovitis, N.; Li, D.; Zhang, N.; Li, J.; Zhang, L.; Xiao, X. Novel approach to energy-efficient flexible job-shop scheduling problems. *Energy* **2022**, *238*, 121773. [CrossRef]
21. Duque, E.T.; Fei, Z.C.; Wang, J.F.; Li, S.Q.; Li, Y.F. Energy Consumption Control of One Machine Manufacturing System with Stochastic Arrivals Based on Fuzzy Logic. In Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 16–19 December 2018; pp. 1503–1507. [CrossRef]
22. Wang, K.; Xue, Y.; Guo, Q.; Shahidehpour, M.; Zhou, Q.; Wang, B.; Sun, H. A Coordinated Reconfiguration Strategy for Multi-Stage Resilience Enhancement in Integrated Power Distribution and Heating Networks. *IEEE Trans. Smart Grid* **2023**. *Early Access*. [CrossRef]
23. Haseeb, K.; Din, I.U.; Almogren, A.; Ahmed, I.; Guizani, M. Intelligent and secure edge-enabled computing model for sustainable cities using green Internet of Things. *Sustain. Cities Soc.* **2021**, *68*, 102779. [CrossRef]

24. Awan, K.A.; Din, I.U.; Almogren, A. A Blockchain-Assisted Trusted Clustering Mechanism for IoT-Enabled Smart Transportation System. *Sustainability* **2022**, *14*, 14889. [CrossRef]
25. Melnyk, S.A.; Ragatz, G.L. Order review/release: Research issues and perspectives. *Int. J. Prod. Res.* **1989**, *27*, 1081–1096. [CrossRef]
26. Oosterman, B.J.; Land, M.J.; Gaalman, G.J.C. The influence of shop characteristics on workload control. *Int. J. Prod. Econ.* **2000**, *68*, 107–119. [CrossRef]
27. Renna, P. Workload control policies under continuous order release. *Prod. Eng. Res. Dev.* **2015**, *9*, 655–664. [CrossRef]
28. Thürer, M.; Qu, T.; Stevenson, M.; Maschek, T.; Godinho Filho, M. Continuous workload control order release revisited: An assessment by simulation. *Int. J. Prod. Res.* **2014**, *52*, 6664–6680. [CrossRef]
29. Twomey, J.; Yildirim, M.; Whitman, B.L.; Liao, H.; Ahmad, J. *Energy Profiles of Manufacturing Equipment for Reducing Energy Consumption in a Production Setting*; Working Paper; Wichita State University: Wichita, KS, USA, 2008.

applied sciences

MDPI

Article

Mixed-Integer Linear Programming, Constraint Programming and a Novel Dedicated Heuristic for Production Scheduling in a Packaging Plant

Soukaina Oujana [1,2,*], Lionel Amodeo [2,*], Farouk Yalaoui [2] and David Brodart [1]

[1] Brodart SAS, 1 Rue du Stand, 10700 Arcis-sur-Aube, France
[2] Laboratory of Computer Science and Digital Society (LIST3N), University of Technology of Troyes, 12 Rue Marie Curie, CS42060, 10004 Troyes Cedex, France
* Correspondence: soukaina.oujana@utt.fr (S.O.); lionel.amodeo@utt.fr (L.A.)

Abstract: In this paper, we are discussing a research project aiming to optimize the scheduling of production orders within a real application in the packaging field. As a first approach, we model the problem as an extended version of the hybrid and flexible flowshop scheduling problem with precedence constraints, parallel machines, and sequence-dependent setups. The optimization objective considered is the minimization of the total tardiness. To tackle this problem, we use two methodologies: mixed-integer linear programming (MILP) and constraint programming (CP). These two models were further extended by adding resource calendar constraints named also availability constraints; this implies that the tasks should be scheduled only when the machine is available. The different proposed models were compared to each other on a set of generated benchmarks that reflect the specific properties of the industrial partner. Finally, as the studied configuration relies on practical real-world application, where thousands of orders are produced monthly, a novel dedicated heuristic was designed to address the need for quick solutions. The latter outperforms the other proposed algorithms for expected total tardiness minimization. The proposed problem can be readily modified to suit a wide range of real-world situations involving the scheduling of activities that share similar characteristics.

Keywords: scheduling; optimization; mixed-integer linear programming; constraint programming; dedicated heuristic; tardiness

1. Introduction

Effective production planning and scheduling attract continuous interest from manufacturing companies, which is a good way to add flexibility to the business, to meet the deadlines promised to the customer, and to ensure the best production efficiency by balancing production needs with available resources, all at minimal cost. From this point of view, the use of robust tools for production scheduling remains a strategic issue because they enable optimizing production and meeting market challenges.

Scheduling is the operational organization of production in the workshop by deciding the order in which tasks pass through the machines, respecting a certain number of constraints to which the workshop is subjected, and according to optimization criteria considered for decision making. In other words, the schedule can be defined as follows: assign the task 'i' to the machine 'k' at a given time 't' while considering, for example, the operator 'p' equipped with the tool 'o' and the mater 'm'.

Among different workshop configurations, a flowshop scheduling problem (FS) arises in the context of repeated production, where jobs are required to visit the stages in the same order and undergo identical processing operations; in other words, all operations of all tasks go through the machines in the same order. In order to cope with real-world problems, improve the overall capacity, add additional flexibility to the production, and avoid bottlenecks if some operations are too long, it is possible to multiply the number of machines that can perform the same operation. The resulting model is known in the literature as hybrid flowshop (HFS), also called flowshop with parallel machines; it consists of a set of processing stages, in which each stage may have several identical or non-identical machines, with at least one stage having two or more parallel machines. The classical hybrid flowshop assumes that all jobs need to visit all stages in the same order. However, in practice, each job might miss out or skip some stages, which can improve the performance of the model and make it better suited for real-world industrial settings. HFS scheduling problem with stage skipping is also called hybrid flexible flowshop scheduling problem. The configuration under study is described in Figure 1.

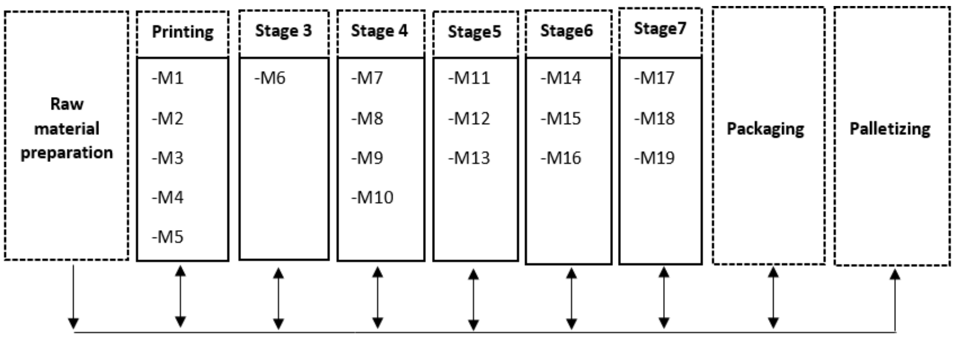


Figure 1. Studied configuration.

In hybrid and flexible flowshop (HFFS) scheduling problems, two decisions should be taken: the assignment of jobs to the parallel machines as well as the sequencing of the jobs allocated to each machine. This problem is known to be NPhard in its simple version and in most of its extensions. As an example, Hoogever et al. [1] demonstrated that preemptive scheduling in a two-stage flowshop with at least two identical parallel machines in one of the stages so as to minimize makespan is NP-hard in the strong sense. Gupta et al. [2] considered a non-preemptive two-stage hybrid flowshop problem in which the first stage contains several identical machines and the second stage contains a single machine; they demonstrated that the problem is NP-hard in the strong sense even when there are only two machines at the first stage. HFFS scheduling problem has been widely applied in various manufacturing environments, and several realistic constraints were considered. A fair amount of research has focused on a variety of realistic constraints, ranging from sequence-dependent setup times, constraint calendar, transportation time, due dates, and so on. Furthermore, several optimization criteria were considered, covering the commonly used makespan, costs, transportation, maximum tardiness and earliness, and the total of tardy job.

In this paper, we make three significant contributions. Firstly, we introduce novel CP and MILP models that take into account specific constraints, including sequence-dependent setups and resource calendar constraints. Secondly, we assess the performance of both models using real industrial benchmarks. Lastly, we propose a dedicated heuristic that effectively addresses the need for fast computation times in practical real-world applications, such as the one studied in this paper, where thousands of orders are produced each month.

The remainder of this paper is organized as follows: Section 2 reviews the state of the art regarding the related papers. The problem description is presented in Section 3. In Section 4, we formulate all the proposed resolution models (MILP, CP, and a novel dedicated heuristic). The computational experiments, allowing to evaluate the performance of the proposed models, are presented in Section 5. Finally, in Section 6, we present the conclusions of our work.

2. State of the Art

The area of flowshop scheduling has been a very active field of research. It was first proposed by Johnson in 1954. Since then, several approaches have been proposed and numerous optimization objectives were considered. The current trends that attracted researchers during the last decade in scheduling problems are toward integrating practical constraints. Among all, we can point out setup time, resource calendar, and machine flexibility.

2.1. Constraints

2.1.1. Setup Constraints

Setup time, also called changeover, is a very important factor in the packaging industry because it may have a significant impact on the overall production cycle. It denotes the required time interval to prepare the necessary material resources. In many real-life situations, a setup often occurs while shifting from one operation to another. Setup time is classified into two categories: sequence-independent setup time and sequence-dependent setup time. Sequence-independent setup time depends solely on the current task regardless of its previous task. Sequence-dependent setup time depends on both the current and immediately preceding task [3,4].

There has been a growth in interest in incorporating setup times in many studies. The main reason why researchers have been motivated to utilize this assumption is to solve scheduling problems in a real manner [5] Liu and Chang [6] addressed the problem of $F_{\mathrm{m}}|St_{sd}, C_{sd}, r_i|\sum ST_i, C_i$. They first formulated the problem as an integer programming problem. Then, they employed a Lagrangian relaxation approach and finally developed a search heuristic. Three major types of heuristics were proposed by Kurz and Askin [7], who explored the $F_{\mathrm{m}}|St_{sd}, C_{sd}, r_i|\sum C_i$ problem, namely insertion heuristics (based on insertion heuristics for the traveling salesman problem), Johnson's algorithm, and a set of naïve greedy heuristics. They investigated these three patterns and identified the range of conditions under which each method performs well. Salmasi et al. [8] proposed a mathematical programming model for $F_{\mathrm{m}}|\mathrm{fmls}, St_{sd}|\sum C_i$ as the problem is proven to be strongly NP-hard; two heuristic algorithms, tabu search (TS) and hybrid ant colony optimization (HACO), were developed to solve the problem. In addition, a lower bounding method based on the branch and price algorithm was developed to assess the performance of the metaheuristic algorithms. An et al. [9] considered the $F_2|\mathrm{wt}, St_{sd}|C_{max}$ problem; they developed several dominance properties, lower bounds, and heuristic algorithms and used the latter to develop an efficient branch and bound algorithm. Cheng et al. [10] tackled the $F_{\mathrm{p}}|St_{sd}|C_{max}$ problem; they proposed a mixed-integer linear programming model to solve small-sized instances. Due to the strong NP hardness of the research problem, an effective metaheuristic, called pairwise iterated greedy (PIG) algorithm, was proposed to solve medium- and large-sized problems. Rossi and Nagano [11] proposed a mixed-integer linear programming (MILP) model for $F_{\mathrm{m}}|St_{sd}|\sum T_i$ problem. They proposed a method to evaluate the total tardiness of a permutation sequence and also introduced a partial acceleration method to calculate the total tardiness in an insertion neighborhood. In addition, they developed a new heuristic to solve the problem efficiently. This heuristic was then integrated into the best metaheuristics available in the literature. Kare and Agrawal [12] studied the $F_{\mathrm{m}}|St_{sd}|\sum wT_i, wE_i$ problem. Three evolutionary metaheuristics were proposed.

2.1.2. Resource Calendar Constraints

Another common and practical constraint found in real environments is to consider the resource calendar. The traditional scheduling problem assumes that machines are continuously available. However, in reality, this is often not the case due to non-availability periods, such as maintenance, vacations, leaves, and so on. Considering these time-off periods for resources is crucial for accurate and realistic scheduling. This helps to determine when resources are available to work on assigned tasks, ensuring that work is only scheduled during available times.

Machine availability constraints encountered in real-world environments can be classified as either fixed or non-fixed [13]. For fixed constraints, the intervals of unavailability are predetermined, whereas they are unknown for the non-fixed constraints. Unavailability periods can also be categorized based on operation preemption as non-preemptive [14], crossable or non-crossable [15], or resumable, semi-resumable, or non-resumable [16]. An operation is known as non-preemptive when its processing on a machine cannot be interrupted until it is totally completed, and after that the concerning machine switches to another operation. An operation interrupted by an unavailability period is called resumable when its processing can continue during the next availability period. It is called non-resumable if it has to restart from the beginning when the performing machine is available again. An operation is known as semi-resumable if it has to partially restart during the next available period. There is other terminology introduced by Mauguière et al. [15]. It concerns unavailability periods allowing interruption of operations: crossable and non-crossable unavailability periods. An unavailability period that allows an operation to be interrupted and resumed after the unavailability period is called crossable, while an unavailability period that does not allow the interruption of any operation is known as non-crossable. Figure 2 gives a description for the notation used for interruptible and non-interruptible operations.

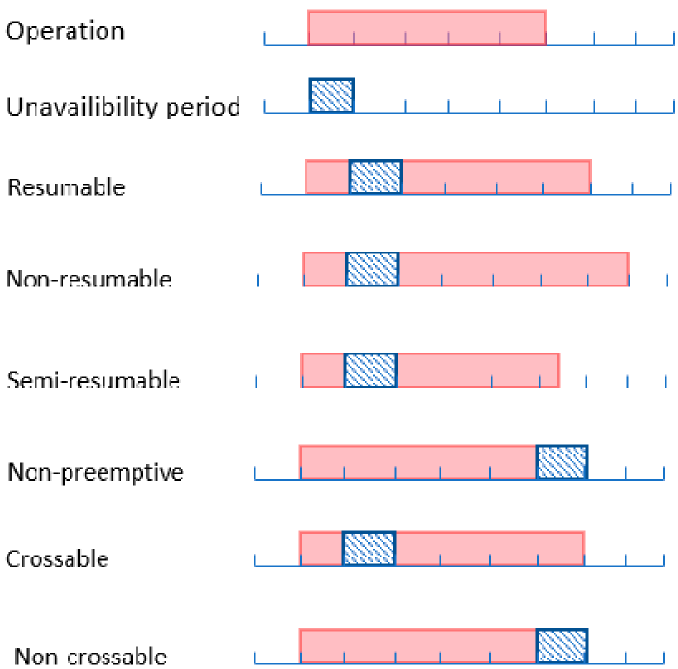


Figure 2. Notations for interruptible/non-interruptible operations.

Bentalleb et al. [17] consider a deterministic case where unavailability periods are known in advance and fixed and correspond to preventive maintenance tasks. They tackled a two-machine job shop scheduling problem with an availability constraint on one machine under makespan minimization. First, two mixed-integer programming (MIP) models were proposed and then some heuristics were performed to solve the problem. Azem et al. [18] investigate the job shop problem where operations can be interrupted by resource unavailability periods. They propose approximation methods based on construction heuristics.

Surprisingly, the literature on the flowshop scheduling problems with resource calendar or fixed machine availability is not abundant. Aggoune et al. [14] address the flowshop scheduling problem with limited machine availability under the makespan criterion and under the assumption that the machines are not available during the whole planning horizon. They propose a heuristic approach based on the geometric approach to approximately solve the problem. Figealska [19] studied the problem of preemptive scheduling in a two-stage flowshop with parallel unrelated machines under makespan minimization. Heuristic algorithms were proposed based on combined linear programming procedures and a genetic algorithm. Laribi et al. [20] investigate an extension of the classical flowshop scheduling problem to the case where jobs processing requires additional nonrenewable resources; the goal is to minimize the makespan. They propose an efficient mathematical model.

2.1.3. Machine Flexibility Constraints

In a modern manufacturing unit, machine flexibility is a very important feature that enables increasing the overall workshop capacities, reducing or eliminating the impact of bottleneck stages and balancing the capacities of the stages for the overall workshop. Such a production unit is characterized by several stages. Each stage is made up of a set of parallel machines. At some stages, the machines are duplicated and a job can be processed on any machine. A flowshop with parallel machines is also known as a multiprocessor flowshop, flexible flowshop, or hybrid flowshop. Machine flexibility has attracted much attention from researchers in recent years. There are several examples provided in the literature, including steelmaking [21,22], industry [23], as well as the semiconductor industry [24,25]. Odugawa et al. [26] provide a survey on several real-world applications, ranging from the metal forming industry to the paper industry to the chemical industry. Some researchers address real-world problems in their papers.

Kochhar et al. [27] exhibit a local search approach to solve highly realistic flexible flow line scheduling with setups, buffer capacities, as well as blocking and breakdowns. Several heuristics are provided by Botta-Genoulaz [28] for the flowshop scheduling problem with multiple identical machines per stage, precedence constraints and time lags, and setups. Ruiz and Maroto [29] provide a metaheuristic, in the form of a genetic algorithm, to a complex generalized flowshop scheduling problem that results from the addition of unrelated parallel machines at each stage, sequence-dependent setup times, as well as machine eligibility. Naderi et al. [30] investigate the problem of hybrid flexible flowshop (HFFS) with sequence-dependent setups, where the objective is to minimize the makespan. They put forward two advanced algorithms that effectively handle the flexible and setup features of this problem. Chen [31] proposed an integer hybrid metaheuristic based on the principles of variable-neighborhood descent and TS for unrelated parallel machines problems with ready times and sequence- and machine-dependent setup times to minimize the weighted number of tardy jobs.

While many papers in the literature have tackled various realistic considerations and constraints, to the best of our knowledge, there has been no effort to jointly address the set of realistic constraints incorporated in the problem formulation of our paper, which include sequence-dependent setups, machine flexibility, and resource calendar constraints.

2.2. Optimization Criteria

Setting the correct optimization criteria or objectives for a scheduling problem is not always an easy task as they are diverse, convoluted, and often conflicting. Plenty of scheduling problems have been studied considering several criteria. The most considered are makespan (C_{max}), total flow time, total tardiness, maximum tardiness, and number of tardy jobs. Makespan and total flow time seek the effective utilization of the manufacturing resources by reducing the elapsed time between the start and the completion of a sequence of operations in a set of machines, while the remaining criteria are related to job due dates. In fact, makespan minimization is significantly important in order to upsurge the utilization of the production system. However, in today's competitive environment, focusing on makespan minimization without meeting the due date is of no use for an industry since meeting customer deadlines is crucial. According to Sen and Gupta [32], when a task is not completed before its due date, some penalties are incurred, such as potential loss of customers, damaged reputation, loss of market competitiveness, penalty clauses if there are any, as well as expediting (the job is assigned quickly to the processing machine at the possible cost of extra setups, double handling of material, inefficient use of workmen and machine), etc. Hence, scheduling problems with tardiness objectives have attracted increasing attention from managers and researchers. Table 1 provides a summary of several significant studies that focus on the tardiness objective. The "Constraints" column contains the various constraints that were taken into account in these studies. The "//m" column refers to parallel machines, the "ST_{sd}" column pertains to sequence-dependent setups, the "d_i" column represents due date constraints, the "w_i" column represents waiting time, and finally the "r_i" column represents release date constraints.

Table 1. Important studies on scheduling problems with tardiness objectives.

Objective Function	Year	Author	Reference	Constraints					Approach
				//m	ST_{sd}	d_i	w_i	r_i	
Total weighted tardiness	1997	Lee and Pinedo	[33]	✓	✓	✓			Dispatching rule ATCS (Apparent Tardiness Cost with Setups)
	2000	Park et al.	[34]	✓		✓			Dispatching rule
	2009	Naderi et al.	[35]		✓		✓		MIP and EMA metaheuristic
	2013	Xi and Jang	[36]		✓			✓	Dispatching rules (ATCS)
	2020	Diana et al.	[37]	✓	✓				VND metaheuristic
Total tardiness	2009	Chen	[31]	✓	✓	✓			Hybrid Approach (ATCS+SA)
	2014	Herr and Goel	[38]		✓	✓			MIP
	2015	Liang et al.	[39]	✓					ACO algorithm
	2018	Lee	[40]	✓					Random iteration greedy metaheuristic
	2020	Rossi and Nagano	[11]		✓				MILP, heuristics and metaheuristics
Makespan and total tardiness/tardy jobs	2009	Naderi et al.	[41]		✓		✓		SA algorithm
	2013	Tran et Ng	[42]				✓		A hybrid water flow algorithm
	2018	Allahverdi et al.	[43]	✓					AA algorithm
	2021	Wan et al.	[44]						A pseudo-polynomial algorithm and a dual FPTAS
	2022	Allali et al.	[45]		✓				MILP and metaheuristics (GA, ABC, MBO)
Tardy jobs	2017	Aydilek et al.	[46]						A DR algorithm
	2019	Najat et al.	[47]	✓					Mathematical programming and heuristics
	2021	Della Croce et al.	[48]	✓					Exponential time approximation algorithms
	2022	Hejl et al.	[49]			✓			A decomposed ILP model
Bi-objective Sum of weighted earliness and weighted tardiness	2008	Behnamian et al.	[50]	✓					A hybrid metaheuristic algorithm that combines ACO, SA, and VNS
	2009	Behnamian et al.	[51]	✓					Three hybrid metaheuristics
	2011	Behnamian et Zandieh	[52]		✓	✓	✓		A discrete colonial competitive algorithm
	2019	Otten et al.	[53]	✓					Heuristic
	2020	Schaller and valente	[54]	✓					BB and heuristics
	2020	Kellerer et al.	[55]						FPTAS

3. Problem Description

The problem under study corresponds to a real industrial problem of a packaging company that prints, on average, 1000 jobs per month. The operations can be categorized into four main groups, ranging from the preparation of printing materials to the printing process and shifting process (winding, perforation, and coating) and finally shaping process aiming to make orders into their final form. Moreover, it is important to point out that this process is characterized by flexibility, where machines might be skipped and not all the machines must be visited by all the jobs.

A job i consists of a number n of operations; each operation O_{ij} can be processed by a subset of machines and has a processing time on machine k, and it may be zero for some jobs as the jobs are not processed in some stages (skipping). Note that p_i denotes the processing time of job i.

Before starting processing, a setup time (ST) is needed between each of two consecutive scheduled jobs on each machine. That is to say that, to transition from the processing of the current operation O_{ij} to the next one $O_{i'j'}$ on machine k, some setup settings must be implemented according to the characteristics of each operation, such as color, size, etc. An example of setup for the printing phase consists of removing the ink colors not required for the next job on that printing machine to free up the ink trays for colors that are required for the next job. Transitioning from one job to another requires to change ink colors. The time required to set up one job for the printing phase can be divided into three steps: the first one to empty the tray from the previous ink, the second one to clean the ink tray, and the last one to reload the appropriate ink color. The global needed setup time depends on the number of color changes. On average, the significant setups may contribute to 40% of the global printing step, including processing time and setup time. However, if a job "i" requires the same color as the previous job "$i-1$", then the setup time for this color may be avoided because the considered printing machine is already loaded with appropriate color and, therefore, major setups are not needed. The setup time duration is correlated to the setup settings' similarities between two consecutive operations. The more resemblance the operations' settings, the shorter the machine setup.

Another important feature of the considered problem is resource calendar constraints (RC), which allow to set the work shifts of all machines. The work shift is a segment of continuous available times of a machine. This means that machines are available only during working times in the calendar. These unavailability periods are the consequence of shift patterns or planned maintenance. On the other hand, the machine setup cannot be interrupted by unavailable periods, and the end of the setup must be immediately followed by the beginning of the operation processing. Furthermore, a transportation time is needed to transport a job from the current processing machine to the next one.

Based on the key features of the considered production system, a production schedule should be planned to maximize the production effectiveness so that the printing line can gain as much production benefit as possible. The production effectiveness can be represented by an objective function that should be defined based on the production targets of the problem. In flexible manufacturing plants operating in a make-to-order environment, the efficient utilization of manufacturing resources is typically pursued to meet delivery deadlines. Thus, in our case, we aim to minimize the total tardiness of all jobs, meaning that we seek to find a job sequence that minimizes the total amount of time by which all jobs are completed after their due dates.

The production problem can be described as a hybrid and flexible flowshop with nineteen unrelated parallel machines, denoted using the classical Graham notation $\mathrm{HFF}_{19}\left|Prec, ST_{sd}, RC, d_i\right|\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{t_i}$ [56]. This classification is based on the features mentioned above, and it is commonly used to represent production systems.

As the first systematic attempt to solve this problem, we construct a mathematical model in the form of mixed-integer linear program (MILP) that considers sequence-dependent setups; we then add waiting constraints and evaluate how it behaves, and, finally, we added resource calendar constraints that enhanced the complexity of the problem. The objective is to both assign jobs to one machine at each stage and then sequence jobs on machines to minimize the total tardiness.

If the completion time of a job is greater than its due date $(C_i > d_i)$, then it is called tardy and tardiness takes positive values. Otherwise, it becomes an early job with a tardiness value equal to zero $(t_i = max(0, C_i - d_i))$.

4. Model and Notations

Job characteristics are modeled as follows:

Let N be the number of jobs to be scheduled. Each job $i(i = 1 \ldots N)$ is composed of a set of operations J_i that must be processed according the defined processing route. Let M be the number of all available material resources "machines". For each operation j, let $m_j \subset M$ be the set of operations that can perform $j \in$ ji and p_{ik} be the corresponding processing times.

To transition from executing operation O_{ij} to operation $\mathrm{O}_{i'j'}$ on machine k, a setup time $\mathrm{st_{iji'j'k}}$ must be incurred. In our problem, the setup time for a job is dependent on the previous job that was processed on the same machine and thus on the job processing sequence. For each machine $K(k = 1 \ldots m)$, let l_k be the number of unavailability periods and $\left[v_k^l, \bar{v}_k^l\right]$ the time window of unavailability of material resource $k \in m$. The processing of each job on the latter can only be preempted by this interval $\left[v_k^l, \bar{v}_k^l\right]$ and resumed once the machine becomes available. Let d_i denote the due date and specify the time limit by which job $i \in N$ should be completed. The number of jobs, their respective processing times, and due dates are predetermined and known beforehand. Each machine has a capacity and can only process one job at a time. A machine can only process one operation at a time. The processing of the latter can be interrupted by an unavailability period. Setups cannot be interrupted by an unavailability period and should occur when the machine is available during the setup interval, and, once completed, the processing of the associated operation should start. There is no limit on the capacity of the intermediate stock (buffer) between the production stages. Finally, the objective is to minimize the total tardiness.

The notation used in this mathematical modelling is summarized in Tables 2 and 3:

Table 2. Notation used for the problem data.

Problem Data	
i, i'	Index for jobs where $i, i' \in \{1, \ldots, N\}$.
j	Index for operations.
O	The total number of operations.
O_{ij}	The j^{th} operation of job $i \in N$.
k	Index for machines where $k \in \{1, \ldots, m\}$.
M	Number of all material resources.
N	Number of jobs to be scheduled.
$\boldsymbol{J_i}$	Set of operations of job $i \in N$.
$\boldsymbol{P_i}$	Processing time job $i \in N$.
$\boldsymbol{d_i}$	Due date of job $i \in N$.
$\boldsymbol{m_j \subset M}$	Set of material resources that can perform the operation $j \in ji$.
$\boldsymbol{St_{iji'j'k}}$	Setup time to pass from the execution of an operation O_j to operation $O_{j'}$ on machine k.
$BigM$	A very large number.
$\boldsymbol{m_{ij} \cap m_{i'j'}}$	Set of machines on which operations j of job i and j' of job i' can be processed.
$\boldsymbol{l_k}$	The number of unavailability periods on machine $k \in m$.
$\boldsymbol{v_k^l}$	The starting time of the lth unavailability period of material resource $k \in m$.
$\boldsymbol{\bar{v}_k^l}$	The ending time of the lth unavailability period of material resource $k \in m$

Table 3. The notation used for the decision variables.

Decision Variables	
X_{ijk} =	1 if the operation O_{ij} is assigned to the material resource k. 0 otherwise.
$Y_{iji'j'k}$ =	1 if the operation O_{ij} is processed before the operation $O_{i'j'}$ on the material resource k. 0 otherwise.
S_{ijk} =	Starting time of the operation O_{ij} on machine k.
C_{ijk} =	Completion time of the operation O_{ij} on machine k.
C_i =	Completion time of job i.

4.1. Mixed-Integer Linear Programming

In this section, the MILP formulation presented in [3] is recalled using the notation of Section 4 and afterwards extended in Section 4.1.2 by adding resource calendar constraints.

4.1.1. Start-Based Model

The start-based model was developed in our previous work [3], with the consideration of sequence-dependent setups, parallel machines, and precedence constraints, and this model is formulated as a mixed-integer linear programming model as below and called $MILP_{wav}$ from now on.

$$\text{Minimize } T = \sum_{i=1}^{n} t_i \tag{1}$$

Subject to:

$$t_i = max(C_i - d_i) \tag{2}$$

$$\sum_{k=1}^{me} X_{ijk} = 1, \forall \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m_j} \tag{3}$$

$$C_{ijk} \geq S_{ijk} + P_{ijk} - BigM\left(1 - X_{ijk}\right), \forall i \in N, j \in Ji, k \in m_j \tag{4}$$

$$\mathrm{S_{ijk}} + \mathrm{C_{ijk}} \leq \mathrm{BigM}(\mathrm{X_{ijk}}\Big), \forall i \in N, j \in Ji, k \in m_j \tag{5}$$

$$C_{ijk} \geq S_{ijk}, \forall i \in N, j \in Ji, k \in m_j \tag{6}$$

$$\sum_{i \in j} \sum_{j,j' \in oi} \sum_{k \in mi} Y_{iji'j'k} = 1, \forall i, i' \in N, j, j' \in J_i, J_{i'} k \in m_{ij} \cap m_{i'j'} \tag{7}$$

$$S_{ijk} \geq C_{i'j'k} + St_{iji'j'k} - BigM\left(1 - Y_{iji'j'k}\right), \forall i, i' \in N, j, j' \in J_i, J_{i'} k \in m_{ij} \cap m_{i'j'} \tag{8}$$

$$S_{i'j'k} \geq C_{ijk} + St_{iji'j'k} - BigM\left(Y_{iji'j'k}\right), \forall i, i' \in N, j, j', J_i, J_{i'}, k \in m_{ij} \cap m_{i'j'} \tag{9}$$

$$C_{ijk} = P_{ijk} + S_{ijk}, \forall i \in N, j \in J_i, k \in m_j \tag{10}$$

$$C_i = \sum_{k=1}^{m_{ij}} C_{ijk}, \forall i \in N, j \in Ji, k \in m_{ij} \tag{11}$$

$$t_i \geq 0, i \in N \tag{12}$$

$$X_{ijk} \in \{0,1\}; \forall i \in N, j \in o_i, \in m_{ij} \tag{13}$$

$$Y_{iji'j'k} \in \{0,1\}, \forall i, i' \in N, j \in J_i, j' \in J_{i'}, k \in m_{ij} \cap m_{i'j'} \tag{14}$$

$$S_{ijk} \geq 0, \forall i \in N, j \in J_i, k \in m_{ij} \tag{15}$$

$$C_{ijk} \geq 0, \forall i \in N, j \in J_i, k \in m_{ij} \tag{16}$$

The objective function (1) aims at minimizing the sum of the total tardiness of all jobs. Constraint (2) provide us with the value of the individual tardiness of each job. Constraint (3) states that each operation can only be assigned to one machine, where the decision variable X_{ijk} is non-zero if operation O_{ij} is assigned to processing unit k and zero otherwise. Constraint (4) ensures that a job's completion time is no earlier than the sum of its start time and processing time. Constraint (5) sets the end date of each job on machines that are not processing the job to 0. Constraint set (6) controls job completion at stages that a job may skip. Constraint set (7) enforces precedence constraints, ensuring that each operation of a job can only begin after its preceding operation has been completed. Constraints (8) and (9) are used together to sequence any pair of tasks (i, i') assigned to the same processing unit k, preventing two jobs from being processed simultaneously on the same machine to ensure the machine is occupied when processing an operation. Constraint set (10) specifies that the completion time of any operation is the sum of its start time and processing time. Constraint set (11) calculates the completion time of a job as the sum of the completion times of all the operations in its processing route. Constraint (12) ensures that only positive tardiness values are considered. Finally, Constraint sets (13) (14), (15), and (16) define the decision variable domains.

4.1.2. Modeling Calendar Constraints

The start-based model is further extended to solve unavailability problems, which are also addressed by incorporating resource calendar constraints. The processing of a job should not start during the time window of unavailability of resource k. That is to say that any operation must be carried out and finished before the arrival of an interval of unavailability. The execution time of an operation must be outside unavailability interval.

$$\left[S_{ijk}, S_{ijk} + P_{ijk}\right] \cap \left[v_k^l, \overset{-}{v}_k^l\right] = \varnothing \tag{17}$$

Constraint (18) allows to calculate the total unavailability period of a machine k that processes job i.

$$u_i = \sum_{k=}^{m} \sum_{l}^{l_k} X_{ik} (\overset{-}{v}_k^l - v_k^l) \tag{18}$$

Constraint (19) calculates the operations completion time

$$C_{ijk} = P_{ijk} + S_{ijk} + St_{iji'j'k} + u_i, \forall i \in N, j \in J_i, k \in m_j \tag{19}$$

Now, we have:

$$C_i = \sum_{k=1}^{mi} P_{ijk} + S_{ijk} + St_{iji'j'k} + u_i, \forall i \in N \tag{20}$$

From now on, we refer to the model that incorporates resource calendar constraints into model MILP_{wav}, MILP_{RC}.

4.2. Constraint Programming

Constraint programming (CP) has good performance and robustness in the optimization field. In fact, it is a strong tool for solving discrete optimization problems; it provides a set of modeling features suitable for a very wide range of complex scheduling problems that do not have a simple formulation. It provides an algebraic language with simple mathematical concepts; commonly, CP framework contains useful structural information; it has the advantage of exposing high declarative, compact, and flexible constraint formulations, which allow us to model the problem correctly and therefore makes it perform well for finding optimal feasible solutions [4].

Here, we have modeled the problem in CP using IBM ILOG CP Optimizer. We will not provide the details of the modeling language used in this paper. For those interested in learning more about this, we recommend referring to [57] and the CP Optimizer reference manual.

4.2.1. Start-Based CP Model

A formulation of the main variables is presented in Table 4 using the concepts of CP Optimizer. From now on, this model is called CP_{wav}.

$$\text{Minimize } T = \textstyle\sum_{i=1}^{n} max(0, t_i) \tag{21}$$

Table 4. CP decision variables.

Decision Variables	
interval β_j =	An interval variable for each operation j
interval α_{jk} =	An optional interval variable for each possible assignment of operation j to machine $k \in m_j$

Subject to:

$$\mathrm{t_i} = \max(0, \mathrm{endOf}(\mathrm{itvs}[\mathrm{C_i}]) - \mathrm{d_i}) \tag{22}$$

$$\mathrm{EndBeforeStart}\left(\beta_j, \left[\alpha_{jk}\right]\right) \quad \mathrm{j} \in \mathrm{Ji},\ \mathrm{k} \in \mathrm{m_i} \tag{23}$$

$$\text{Alternative } \left(\beta_j, \beta_{j'}\right) \ j, j' \in Ji \tag{24}$$

$$\text{noOverlap}\left(\left[\alpha_{jk}\right]\right) \quad j \in J, \forall \mathrm{k} \in \mathrm{m_j} \tag{25}$$

$$d_i \geq \beta_i.\text{end } \forall \mathrm{i} \in \mathrm{N} \tag{26}$$

$$\text{interval } \alpha_{jk}, \text{opt}, \text{size} = P_{jk} + St_{jj'k}, i \in V, k \in m_i \tag{27}$$

$$\text{endAtStart}\left(St_{jj'k}, \alpha_{jk}\right) j \in J, \forall \mathrm{k} \in \mathrm{m_j} \tag{28}$$

The objective function is to minimize the total tardiness (21), given by the difference between the job's end value and due date (22). The EndBeforeStart constraints (23) represent the precedence constraints between interval variables. Alternative constraints (24) represent the assignment constraints stating that each operation must be performed on exactly one machine. Constraint (25) defines the nonoverlapping constraint; that is to say that, during the interval $\left[\alpha_{jk}\right]$, which represents the assignment of an operation j to machine *k*, the latter cannot overlap; e.g., the machine is busy during this interval.

The constraint endAtStart (α, β) is used to state that the end of a given interval variable α, equals the start of a given interval variable β. We use this constraint (28) to ensure that the end of a setup should be followed by the execution of the considered operation.

4.2.2. Modeling Calendar Constraints

The considered processing line is periodically submitted to calendar constraints; this means that machines are not available during the whole planning horizon. To consider machines' unavailability, variable α_{jk} should be modulated by adding an intensity step function F_k that represents the unavailability interval of machine k. In CP optimizer, Intensity is a stepwise function that applies a measure of usage or utility over an interval length. The intensity is 0% during the unavailability interval $\left[\upsilon_k^l, \bar{\upsilon}_k^l\right]$ and 100% outside this interval. Therefore, modelling machines' unavailability can simply be formulated by constraint (29)

$$\text{interval } \alpha_{jk}, \text{opt}, \text{size} = \mathrm{P_{jk}}, \text{intesity} = \mathrm{F_k}, \mathrm{j} \in \mathrm{J}, \forall \mathrm{k} \in \mathrm{m_j} \tag{29}$$

An additional feature of our problem is that the setup cannot occur during an unavailability period. To model this feature, we use the predefined constraint forbidExtent (a,U). This expression prevents an interval variable from being scheduled during any time point within the augmented horizon that is not also within one of the disjoint time windows.

$$\text{forbidExtent}\left(St_{ii'k}, F_k\right), \quad i, i' \in N, \forall \mathrm{k} \in \mathrm{m_i} \tag{30}$$

Forbidden start constraint forbidStart(α, F) states that, whenever the interval is present, it cannot start at a value t where $F(t) = 0$. In the same sense, Forbidden end constraint forbidEnd(α, F) states that, whenever the interval is present, it cannot end at a value t where $F(t-1) = 0$. We use constraints (31) and (2) to ensure the respect of unavailability periods.

$$\text{forbidStart}\left(\alpha_{jk}, F_k\right), j \in J, \forall \mathrm{k} \in \mathrm{m_i} \tag{31}$$

$$\text{forbidEnd}\left(\alpha_{jk}, F_k\right), j \in J, \forall \mathrm{k} \in \mathrm{m_i}$$

From now on, we refer to the model that incorporates resource calendar constraints into model $\mathrm{CP_{wav}}$, $\mathrm{CP_{RC}}$.

4.3. Dedicated Heuristic

In order to meet the needs of an industrial environment, we need to be able to develop quick and effective solutions that can be used to solve the various tasks that are involved for a real industrial framework. Unfortunately, the exact resolution approaches presented previously cannot sufficiently address the requirements for real industrial-size instances (more than 100 tasks). A very common difficulty when trying to solve such large-sized instances with the MILP model is running out of memory. The CP model reaches better solutions in a short time, but, similarly, the solver has some issues regarding the instances' size. In this section, we propose an effective dedicated heuristic that performs well and finds good-quality solutions within a reasonable amount of time.

This heuristic follows the logic of a greedy algorithm, which is a type of problem solving technique that involves making a series of decisions in order to find the best solution. It works by making the best decision at each step without considering the long-term consequences of the decisions. The algorithm works by considering the most immediate benefit of each decision and choosing the one that provides the lowest tardiness. This procedure is repeated until all jobs have been inserted, resulting in a complete candidate solution.

The main steps of this dedicated heuristic are given below:

Step 1. Find earliest schedule
Step 2. Check machine's busyness
Step 3. Setting operation's schedule

This heuristic was coded on python. The detailed procedure of the heuristic is presented in Appendix A.

5. Experimental Results

5.1. Performance of MILP and CP Models

In this section, the performance of the proposed models is evaluated. We use ILOG Cplex 12.10 software and CP Optimizer (CPO) for solving the MILP model and the CP model, respectively, using a DELL personal computer equipped with an Intel® Core™ i5-8250U @ 1.6 1.8 GHz CPU, 8 GB RAM, and Window 10 operating system.

This section begins with a description of the numerical instances that were tested. Then, the different results tables are presented and, at the end, comparisons between the different algorithms are made.

5.1.1. Test Instances

To validate the proposed approaches in this work, we present in this section a description of the test instances that were used. Most of the datasets were initialized on the real database of the studied printing company over a period of 2 weeks. We collected from the production database all the data related to products: operations, processing times, setup times, waiting times, and resource calendar.

To test the performance behavior of the proposed solution approaches and to investigate their efficiency, it is necessary to build several sets of instances in various production environments and different conditions. To this end, some test problems have been applied in a variety of conditions with inspiration from the illustrated case study. Each test set is generated by varying the problem size. It can be characterized by N, the number of jobs, O, the total number of operations, and M, the number of machines. The result tables will not mention the number of machines as it remains constant at 19. The different instances are named WOS for Workshop Scheduling followed by the number of the instance.

An extensive set of numerical experiments have been conducted by considering different problem sizes. The aim is to investigate which jobs and operations the model is not able to find solutions for in a reasonable resolution time.

Based on the combination of the two abovementioned factors, two categories of instances are arranged as the small- and large-sized instances. These categories correspond to different workload situations, respectively: low-workload situation and normal- to high-workload situation.

5.1.2. Experimental Results

In this section, we intend to evaluate the proposed models.

We set the stopping criteria parameters as follows: the time limit CPU is equal to 30 min and the maximum iterations equal to 1000. The performances of the models are evaluated thanks to real data of the workshop. Test results are discussed below.

Several set instance sets have been created with N ranging from

- {5, 8, 10, 13, 15 to 20} for small-sized instances.
- {30, 40, 50, 65, 70, 75, 80 to 100} for large-sized instances

For each set, at least two test instances were generated by varying the number of operations. For each problem class, an effort measurement is completed by calculating the associated total tardiness and the required CPU time. The optimal values that are obtained for tardiness have been distinguished with bold numbers. When applying both formulations to the test instances, a total of 160 experiments were carried out.

Instances without Resource Calendar Constraints

We now present some results on the solution quality obtained with the different models that do not take into account resource calendar constraints.

1. Small-Sized Instances

In this subsection, the general performance of the MILP and CP models is evaluated by a set of small-sized instances. Several instance sets have been created with n ranging from 5 to 20. Table 5 provides an overview of the obtained results. For each problem, the name, the number of jobs, the number of operations and machines, the total tardiness T in minutes, and the solution time in seconds are shown for both models ($\mathrm{MILP_{wav}}$ and $\mathrm{CP_{wav}}$).

Table 5. Main characteristics of the considered small-sized instances and comparison of $MILP_{wav}$ and CP_{wav} models.

Instance Characteristics			$MILP_{wav}$		CP_{wav}	
Instance	**N**	**O**	T_{MILP}	CPU_{MILP}	T_{CP}	CPU_{CP}
WOS1	5	7	**0**	5	**0**	4
WOS2	5	12	**0**	8	**0**	4
WOS3	5	20	**0**	12	**0**	4
WOS4	8	10	**0**	11	**0**	4
WOS5	8	25	**0**	50	**0**	5
WOS6	8	30	**0**	69	**0**	5
WOS7	10	17	**0**	58	**0**	5
WOS8	10	29	**0**	110	**0**	5
WOS9	10	43	**0**	180	**0**	12
WOS10	10	49	**0**	270	**0**	12
WOS11	13	18	**0**	90	**0**	12
WOS12	13	34	**0**	250	**0**	21
WOS13	13	49	**0**	360	**0**	21
WOS14	15	20	**2870**	240	**2870**	21
WOS15	15	45	**4076**	300	**4076**	32
WOS16	15	53	**5760**	410	**5760**	32
WOS17	15	59	**7120**	360	**7120**	32
WOS18	20	29	**8200**	380	**8200**	45
WOS19	20	55	**9590**	520	**9590**	40
WOS20	20	64	**14,200**	730	**14,200**	45
Average	12	33	2591	221	2591	18

Table 5. *Cont.*

Optimal values in bold.

As the results show, the MILP model provides a great performance; it is capable of solving to optimality all the small-sized problems up to n = 20 and o = 64 within a reasonable time. The CP model, on the other hand, seems to be performing better regarding the resolution time. For all the studied instances, the MILP model took longer to achieve an optimal solution. Figure 3 provides a time comparison between the resolution of the CP and MILP models. We can clearly see that the resolution time difference becomes more noticeable as the number of jobs increases.

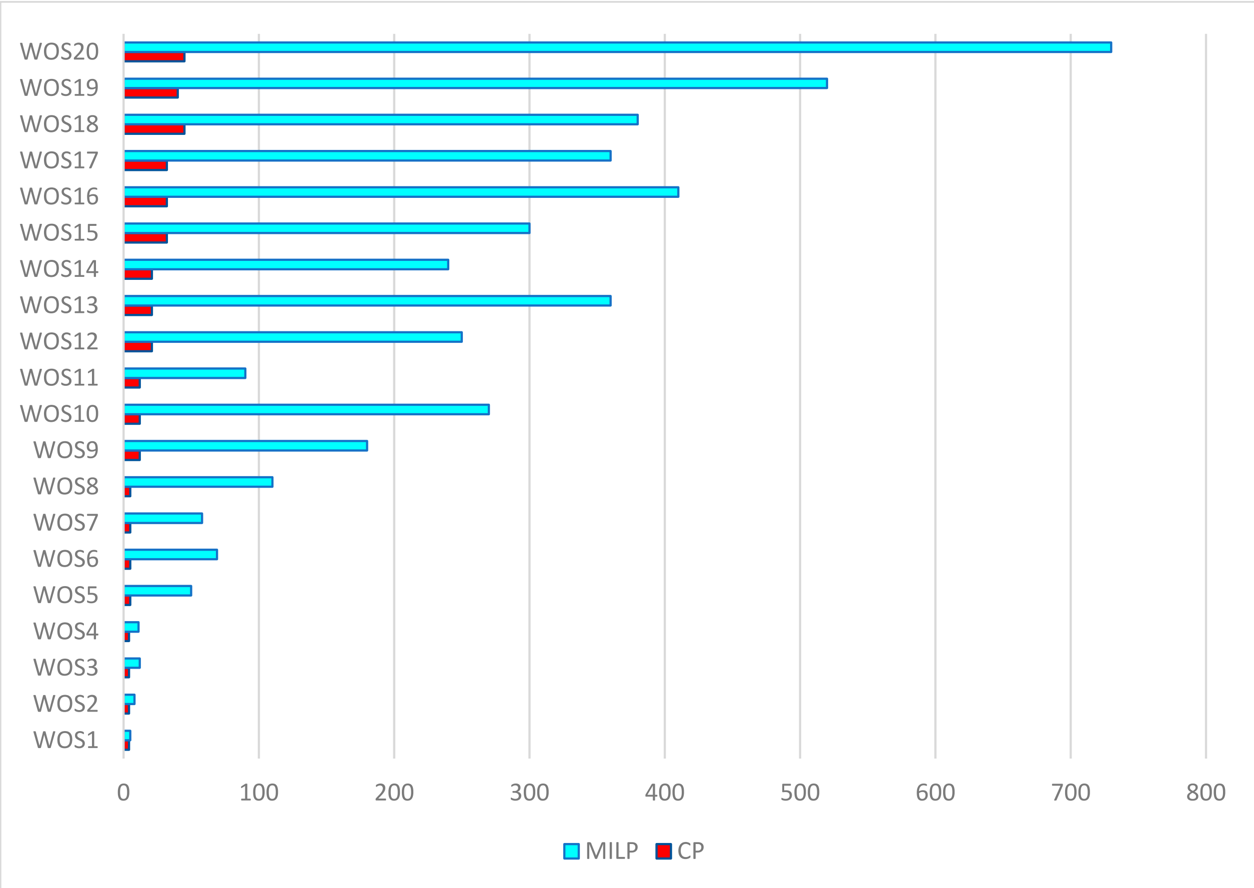


Figure 3. Resolution time (CPU) comparison between CP and MILP models.

2. Large-Sized Problems

To further validate the performance of the proposed models, larger-sized instances are evaluated. Table 6 summarizes the corresponding computational results.

Table 6. Main characteristics of the considered large-sized instances and comparison of $MILP_{wav}$ and CP_{wav} models.

Instance Characteristics			$MILP_{wav}$		CP_{wav}	
Instance	**N**	**O**	T_{MILP}	CPU_{MILP}	T_{CP}	CPU_{CP}
LWOS1	30	66	**0**	320	**0**	6
LWOS2	30	80	**5760**	850	**5760**	6
LWOS3	40	88	9852	1710	**9712**	6
LWOS4	40	96	_	>1800	**9980**	6
LWOS5	50	110	_	>1800	**12,100**	12
LWOS6	50	127	_	>1800	**19,560**	12
LWOS7	65	143	_	>1800	**19,800**	12
LWOS8	65	150	_	>1800	**21,600**	30
LWOS9	65	165	_	>1800	**22,400**	26
LWOS10	65	185	_	>1800	**23,980**	26
LWOS11	70	164	_	>1800	**23,800**	30
LWOS12	70	172	_	>1800	**25,000**	42
LWOS13	70	190	_	>1800	**26,960**	42
LWOS14	75	182	_	>1800	**26,740**	73
LWOS15	75	198	_	>1800	**27,880**	73
LWOS16	75	212	_	>1800	**29,660**	73
LWOS17	80	210	_	>1800	**29,920**	49
LWOS18	80	225	_	>1800	**38,500**	70
LWOS19	100	320	_	>1800	**40,940**	87
LWOS20	100	380	_	>1800	**48,520**	87
Average	65	173	_	_	**23,141**	38

Optimal values in bold.

As can be observed, up to N = 40 and O = 88, the MIP model is unable to find a solution within 1800 s, whereas the CP model still finds an optimal solution for all instances in 38 s on average. The first conclusion that can be drawn is that CP is much faster than MILP. This experimentation confirms CP's outstanding performance for the problem under study.

Instances with Resource Calendar Constraints

This subsection shows the results of the instances on the models that incorporate resource calendar constraints.

1. Small-Sized Instances

The results of the computational comparison for each combination of n and m are presented in Table 7.

Table 7. Main characteristics of the considered small-sized instances and comparison of $MILP_{RC}$ and CP_{RC} models.

Instance Characteristics				$MILP_{RC}$		CP_{RC}	
Instance	**N**	**O**	**U**	T_{MILP}	CPU_{MILP}	T_{CP}	CPU_{CP}
RCWOS1	5	7	1	**0**	10	**0**	302
RCWOS2	5	12	1	**0**	14	**0**	302
RCWOS3	5	20	2	**0**	18	**0**	302
RCWOS4	8	10	2	**0**	15	**0**	302
RCWOS5	8	25	2	**0**	58	**0**	302
RCWOS6	8	30	3	**0**	82	**0**	950

Table 7. *Cont.*

Instance Characteristics				$MILP_{RC}$		CP_{RC}	
Instance	**N**	**O**	**U**	T_{MILP}	CPU_{MILP}	T_{CP}	CPU_{CP}
RCWOS7	10	17	3	**0**	68	**0**	950
RCWOS8	10	29	3	**0**	140	**0**	950
RCWOS9	10	43	3	**180**	240	**180**	950
RCWOS10	10	49	4	**1330**	320	**1330**	950
RCWOS11	13	18	4	**685**	40	**685**	950
RCWOS12	13	34	4	**1258**	380	**1258**	950
RCWOS13	13	49	4	**2780**	450	**2780**	1100
RCWOS14	15	20	4	**4200**	490	**4200**	1100
RCWOS15	15	45	5	**7200**	820	**7200**	1100
RCWOS16	15	53	5	8400	1080	**8320**	1100
RCWOS17	15	59	5	9600	1202	**8592**	1300
RCWOS18	20	29	5	10,800	1440	**9987**	1300
RCWOS19	20	55	6	_	>1800	**12,600**	1300
RCWOS20	20	64	7	_	>1800	**18,600**	1300
Average	12	33	4	2609	326	3783	888

Optimal values in bold.

If we analyze the results when solving the $MILP_{RC}$ and CP_{RC} models with small-sized instances that consider resource calendar constraints, the presence of a high number of unavailability periods decreases, even more regarding the performance of the MILP model (the model only obtains 15 out of 20 optimal solutions and 20 of 20 feasible solutions). The CP model, on the other hand, seems to perform well and is still able to obtain optimal solutions even when the number of unavailabilities is high.

2. Medium- and Large-Sized Instances

The computational results for the medium/large-sized problems are summarized in the Table 8 below.

Table 8. Main characteristics of the considered large-sized instances and comparison of $MILP_{RC}$ and CP_{RC} model.

Instance Characteristics				$MILP_{RC}$		CP_{RC}	
Instance	**N**	**O**	**U**	T_{MILP}	CPU_{MILP}	T_{CP}	CPU_{CP}
LRCWOS1	30	66	1	_	>1800	**0**	1300
LRCWOS2	30	80	1	_	>1800	**6760**	1300
LRCWOS3	40	88	2	_	>1800	**9900**	1300
LRCWOS4	40	96	2	_	>1800	**9998**	1300
LRCWOS5	50	110	2	_	>1800	13,100	1300
LRCWOS6	50	127	3	_	>1800	19,760	1300
LRCWOS7	65	143	3	_	>1800	20,100	1487
LRCWOS8	65	150	3	_	>1800	21,900	1487
LRCWOS9	65	165	3	_	>1800	23,254	1487
LRCWOS10	65	185	4	_	>1800	23,978	1487
LRCWOS11	70	164	4	_	>1800	23,900	1487
LRCWOS12	70	172	4	_	>1800	26,020	1487
LRCWOS13	70	190	4	_	>1800	26,990	1580
LRCWOS14	75	182	4	_	>1800	26,840	1580
LRCWOS15	75	198	5	_	>1800	28,520	1580
LRCWOS16	75	212	5	_	>1800	29,760	1580
LRCWOS17	80	210	5	_	>1800	_	>1800
LRCWOS18	80	225	5	_	>1800	_	>1800
LRCWOS19	100	320	6	_	>1800	_	>1800
LRCWOS20	100	380	7	_	>1800	_	>1800
Average	65	173	4	_	_	19,424	_

Optimal values in bold.

The experimental results show that the $MILP_{RC}$ model is not able to solve any problem of size up to 30 jobs, while the CP_{RC} model solved a large number of instances up to 75 jobs.

5.1.3. Discussion

According to the results of the experiments, the CP algorithm is more efficient than the MILP model when it comes to solving our scheduling problem for both *wav* (without ressource calendar constraints) and *RC* formulations (with resource calendar constraints). It can perform well in handling any size of problem and proves the optimality of a large number of instances. Even with high-availability periods, the CP model can still find optimal solutions. It also proves the optimality of several instances and outperforms the MILP model when it comes to finding feasible solutions.

Summing up, we can clearly see that the computational effort required to solve our scheduling problem depends on the size of instances and the number of unavailability periods. The difference between CP and MILP increases as the number of jobs and the number of unavailability periods increase. CP can provide significant savings in computational effort compared to MILP formulation and finds better solutions and is the best overall in all instances.

5.2. Dedicated Heuristic

For testing the performance of the proposed dedicated heuristic method, we generated a benchmark composed of several sets of instances with different problem sizes by using the real data obtained from the manufacturing environment of the plant. Accordingly, there are 10 groups of benchmark problems of different sizes, varying from 60 to 150 jobs.

Table 9 provides for each instance the tardiness found, denoted by T_{Dh}, as well as the execution time (CPUs column) to reach the best value. The column denoted by T_{real} recalls the real results obtained by the planner. Finally, the gap between both solutions is calculated in the column (gap). The last row represents the average values. The values denoted in bold indicate that the heuristic reaches the optimal value for the considered instances, meaning that the solution found by the heuristic is equal to the one obtained by the exact method "CP". For instances up to 80 tasks, the results obtained with the MILP and CP models are not provided since the solver ran out of memory before providing any initial solution.

Table 9. Main characteristics of the considered large-sized instances and comparison of $MILP_{RC}$ and CP_{RC} models.

Instance Characteristics							
N	**O**	**M**	**U**	T_{real}	T_{Dh}	CPU_{DH}	**Gap**
60	148	19	5	30,120	**22,695**	4	25%
70	160	19	6	48,215	**27,458**	4	43%
80	189	19	7	68,743	**38,548**	6	44%
90	210	19	9	80,471	49,895	8	38%
100	260	19	9	94,875	58,951	8	38%
110	298	19	11	100,458	64,251	8	36%
120	352	19	12	124,524	70,589	12	43%
130	397	19	15	150,427	86,758	12	42%
140	410	19	16	159,751	89,827	12	44%
150	480	19	20	180,058	118,745	19	34%
105	290	19	11	103,764	62,772	9	39%

Optimal values in bold.

According to Table 9, if we compare the results of the dedicated heuristic against the real results obtained by the planner, we see that the tardiness obtained by the heuristic is significantly lower than that obtained by the planner, with an average gap of 39%. On average, the dedicated heuristic provides a better solution overall for all the tested instances within a reasonable time compared to the real solution proposed by the planner, which proves the efficiency of the dedicated heuristic.

6. Conclusions

This paper aims to apply operations research techniques to schedule activities within a packaging company. It examines a difficult scheduling problem, which involves a hybrid and flexible flowshop with various challenging features, such as parallel machines, precedence constraints, sequence-dependent setup times, and resource calendar constraints. The paper presents and analyzes two solutions for the problem using MILP and CP Optimizer. MILP is a general-purpose solver, while CP Optimizer is specifically designed for scheduling problems and has its own modeling language. The study compares the effectiveness of the IBM ILOG CPLEX MILP and IBM ILOG CP Optimizer solvers based on their ability to handle realistic problem sizes, with some showing promise on small instances but struggling on larger ones. From the foregoing, MILP formulation performed well for small-sized instances but struggled to find solutions for large-sized instances, or ones with a high proportion of unavailability periods. The CP formulation performed better for large-sized instances and ones with a high proportion of unavailability periods. Therefore, CP Optimizer is more successful in finding optimal solutions for a greater number of instances than MILP. To deal with large-sized instances, a dedicated heuristic was also proposed to provide good-quality solutions in reduced time. Thus, this heuristic is mainly recommended for large-size problems. Future work should focus on improving the proposed algorithm by adding some dispatching rules and investigating a comparable method for resolving scheduling issues with restricted availability, where operations may be suspended due to availability periods and resumed later, with or without incurring penalties.

Author Contributions: Writing—original draft, S.O.; Writing—review & editing, L.A. and F.Y.; Supervision, D.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by BRODART SAS and financially supported by the National Association of Technical Research (NATR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality of the company.

Acknowledgments: The authors would like to express their gratitude to the editorial board of the journal as well as the anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Pseudo algorithm of the dedicated heuristic.

Pseudo-Algorithm

```
*****************
SCHEDULING METHODS
*****************
Determine an operation's schedule
---------------------------------
    a. initialize
        - l = operation's processing time (setup+execution)
        - ls = operation's setup time
        - s = start
        - ss = None, the actual start, es = None, the start of the execution, r = None, the available time
        - ee = s, the end of the execution
        - A = machines' availabilities (list of int couples representing each an availability window)
```

Table A1. *Cont.*

Pseudo-Algorithm
b. iteration - b = 0, the availability bucket - While l>0 and b<∣A∣ (we still processing time and availability buckets • B = A[b], B is the current availability bucket • si = B[0] (interval start), ei = B[1] (interval end) • if ei<=ee (if this intervals ends before the moving counter ee) ○ continue to next interval c. Set the availability time, r = ee + operation's waiting time Find earliest schedule ---------------------- a. Try to schedule at time - determine a timing from time timing = determineTiming() - check if the machine is busy any time between timing.start and timing.end busy = checkBusy() - if not(busy) • return timing and end b. Else, try to schedule at each busyness interval's end for [si,ei] in the machine's busyness intervals - if ei<time => skip and continue to next interval - timing = determine a timing from ei - busy = check if machine is busy in that timing - if not(busy) • return timing and end Check machine's busyness ------------------------ Setting operation's schedule ---------------------------- a. Set the operation's attribute (start,exec,end,available,machine) to (timing[1],timing[2],timing[3],timing[4],machine.id) b. Add the interval [timing[1],timing[3]] is the machine's busyness and reorder the busyness intervals by increasing values c. Find the next operation nextOp in this operation's parent job d. If nextOp exists, set its release time to timing[4]

References

1. Hoogeveen, J.A.; Lenstra, J.K.; Veltman, B. Preemptive Scheduling in a Two-Stage Multiprocessor Flow Shop Is NP-Hard. *Eur. J. Oper. Res.* **1996**, *89*, 172–175. [CrossRef]
2. Gupta, J.N.D.; Hariri, A.M.A.; Potts, C.N. Scheduling a Two-Stage Hybrid Flow Shop with Parallel Machines at the First Stage. *Ann. Oper. Res.* **1997**, *69*, 171–191. [CrossRef]
3. Oujana, S.; Yalaoui, F.; Amodeo, L. A Linear Programming Approach for Hybrid Flexible Flow Shop with Sequencedependent Setup Times to Minimise Total Tardiness. In Proceedings of the 17th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2021), Budapest, Hungary, 7–9 June 2021.
4. Oujana, S.; Amodeo, L.; Yalaoui, F.; Brodart, D. Solving a Realistic Hybrid and Flexible Flow Shop Scheduling Problem through Constraint Programming: Industrial Case in a Packaging Company. In Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022; Volume 1, pp. 106–111.
5. Naderi, B.; Gohari, S.; Yazdani, M. Hybrid Flexible Flowshop Problems: Models and Solution Methods. *Appl. Math. Model.* **2014**, *38*, 5767–5780. [CrossRef]
6. Liu, C.-Y.; Chang, S.-C. Scheduling Flexible Flow Shops with Sequence-Dependent Setup Effects. *IEEE Trans. Robot. Autom.* **2000**, *16*, 408–419. [CrossRef]
7. Kurz, M.E.; Askin, R.G. Comparing Scheduling Rules for Flexible Flow Lines. *Int. J. Prod. Econ.* **2003**, *85*, 371–388. [CrossRef]
8. Salmasi, N.; Logendran, R.; Skandari, M.R. Total Flow Time Minimization in a Flowshop Sequence-Dependent Group Scheduling Problem. *Comput. Oper. Res.* **2010**, *37*, 199–212. [CrossRef]
9. An, Y.-J.; Kim, Y.-D.; Choi, S.-W. Minimizing Makespan in a Two-Machine Flowshop with a Limited Waiting Time Constraint and Sequence-Dependent Setup Times. *Comput. Oper. Res.* **2016**, *71*, 127–136. [CrossRef]
10. Cheng, C.-Y.; Ying, K.-C.; Li, S.-F.; Hsieh, Y.-C. Minimizing Makespan in Mixed No-Wait Flowshops with Sequence-Dependent Setup Times. *Comput. Ind. Eng.* **2019**, *130*, 338–347. [CrossRef]
11. Rossi, F.L.; Nagano, M.S. Heuristics and Metaheuristics for the Mixed No-Idle Flowshop with Sequence-Dependent Setup Times and Total Tardiness Minimisation. *Swarm Evol. Comput.* **2020**, *55*, 100689. [CrossRef]
12. Khare, A.; Agrawal, S. Scheduling Hybrid Flowshop with Sequence-Dependent Setup Times and Due Windows to Minimize Total Weighted Earliness and Tardiness. *Comput. Ind. Eng.* **2019**, *135*, 780–792. [CrossRef]

13. Mati, Y. Minimizing the Makespan in the Non-Preemptive Job-Shop Scheduling with Limited Machine Availability. *Comput. Ind. Eng.* **2010**, *59*, 537–543. [CrossRef]
14. Aggoune, R.; Portmann, M.-C. Flow Shop Scheduling Problem with Limited Machine Availability: A Heuristic Approach. *Int. J. Prod. Econ.* **2006**, *99*, 4–15. [CrossRef]
15. Mauguière, P.; Billaut, J.-C.; Bouquard, J.-L. New Single Machine and Job-Shop Scheduling Problems with Availability Constraints. *J. Sched.* **2005**, *8*, 211–231. [CrossRef]
16. Lee, C.-Y. Minimizing the Makespan in the Two-Machine Flowshop Scheduling Problem with an Availability Constraint. *Oper. Res. Lett.* **1997**, *20*, 129–139. [CrossRef]
17. Benttaleb, M.; Hnaien, F.; Yalaoui, F. Two-Machine Job Shop Problem for Makespan Minimization under Availability Constraint. *IFAC Pap.* **2016**, *49*, 132–137. [CrossRef]
18. Azem, S.; Aggoune, R.; Dauzère-Pérès, S. Heuristics for Job Shop Scheduling with Limited Machine Availability. *IFAC Proc. Vol.* **2012**, *45*, 1395–1400. [CrossRef]
19. Figielska, E. Heuristic algorithms for scheduling in a flowshop with resource constraints. *IFAC Proc. Vol.* **2007**, *40*, 325–330. [CrossRef]
20. Laribi, I.; Yalaoui, F.; Belkaid, F.; Sari, Z. Heuristics for Solving Flow Shop Scheduling Problem under Resources Constraints. *IFAC Pap.* **2016**, *49*, 1478–1483. [CrossRef]
21. Pan, Q.-K.; Wang, L.; Mao, K.; Zhao, J.-H.; Zhang, M. An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 307–322. [CrossRef]
22. Long, J.; Zheng, Z.; Gao, X.; Pardalos, P.M. Scheduling a Realistic Hybrid Flow Shop with Stage Skipping and Adjustable Processing Time in Steel Plants. *Appl. Soft Comput.* **2018**, *64*, 536–549. [CrossRef]
23. Koch, C.; Arbaoui, T.; Ouazene, Y.; Yalaoui, F.; De Brunier, H.; Jaunet, N.; De Wulf, A. A Matheuristic Approach for Solving a Simultaneous Lot Sizing and Scheduling Problem with Client Prioritization in Tire Industry. *Comput. Ind. Eng.* **2022**, *165*, 107932. [CrossRef]
24. Quadt, D.; Kuhn*, H. Conceptual Framework for Lot-Sizing and Scheduling of Flexible Flow Lines. *Int. J. Prod. Res.* **2005**, *43*, 2291–2308. [CrossRef]
25. Quadt, D.; Kuhn, H. Capacitated Lot-Sizing and Scheduling with Parallel Machines, Back-Orders, and Setup Carry-Over. *Nav. Res. Logist. NRL* **2009**, *56*, 366–384. [CrossRef]
26. Oduguwa, V.; Tiwari, A.; Roy, R. Evolutionary Computing in Manufacturing Industry: An Overview of Recent Applications. *Appl. Soft Comput. J.* **2005**, *5*, 281–299. [CrossRef]
27. Kochhar, S.; Morris, R.J.T.; Wong, W.S. The Local Search Approach to Flexible Flow Line Scheduling. *Eng. Costs Prod. Econ.* **1988**, *14*, 25–37. [CrossRef]
28. Botta-Genoulaz, V. Hybrid Flow Shop Scheduling with Precedence Constraints and Time Lags to Minimize Maximum Lateness. *Int. J. Prod. Econ.* **2000**, *64*, 101–111. [CrossRef]
29. Ruiz, R.; Maroto, C. A Genetic Algorithm for Hybrid Flowshops with Sequence Dependent Setup Times and Machine Eligibility. *Eur. J. Oper. Res.* **2006**, *169*, 781–800. [CrossRef]
30. Naderi, B.; Ruiz, R.; Zandieh, M. Algorithms for a Realistic Variant of Flowshop Scheduling. *Comput. Oper. Res.* **2010**, *37*, 236–246. [CrossRef]
31. Chen, J.-F. Scheduling on Unrelated Parallel Machines with Sequence- and Machine-Dependent Setup Times and Due-Date Constraints. *Int. J. Adv. Manuf. Technol.* **2009**, *44*, 1204–1212. [CrossRef]
32. Sen, T.; Gupta, S.K. A State-of-Art Survey of Static Scheduling Research Involving Due Dates. *Omega* **1984**, *12*, 63–76. [CrossRef]
33. Lee, Y.H.; Pinedo, M. Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times. *Eur. J. Oper. Res.* **1997**, *100*, 464–474. [CrossRef]
34. Park, Y.; Kim, S.; Lee, Y. Scheduling Jobs on Parallel Machines Applying Neural Network and Heuristic Rules. *Comput. Ind. Eng.* **2000**, *38*, 189–202. [CrossRef]
35. Naderi, B.; Zandieh, M.; Shirazi, M.A.H.A. Modeling and Scheduling a Case of Flexible Flowshops: Total Weighted Tardiness Minimization. *Comput. Ind. Eng.* **2009**, *57*, 1258–1267. [CrossRef]
36. Xi, Y.; Jang, J. Minimizing Total Weighted Tardiness on a Single Machine with Sequence-Dependent Setup and Future Ready Time. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 281–294. [CrossRef]
37. Diana, R.; Souza, S. Analysis of Variable Neighborhood Descent as a Local Search Operator for Total Weighted Tardiness Problem on Unrelated Parallel Machines. *Comput. Oper. Res.* **2020**, *117*, 104886. [CrossRef]
38. Herr, O.; Goel, A. Comparison of Two Integer Programming Formulations for a Single Machine Family Scheduling Problem to Minimize Total Tardiness. *Procedia CIRP* **2014**, *19*, 174–179. [CrossRef]
39. Liang, P.; Yang, H.; Liu, G.; Guo, J. An Ant Optimization Model for Unrelated Parallel Machine Scheduling with Energy Consumption and Total Tardiness. *Math. Probl. Eng.* **2015**, *2015*, e907034. [CrossRef]
40. Lee, C.-H. A Dispatching Rule and a Random Iterated Greedy Metaheuristic for Identical Parallel Machine Scheduling to Minimize Total Tardiness. *Int. J. Prod. Res.* **2018**, *56*, 2292–2308. [CrossRef]
41. Naderi, B.; Zandieh, M.; Khaleghi Ghoshe Balagh, A.; Roshanaei, V. An Improved Simulated Annealing for Hybrid Flowshops with Sequence-Dependent Setup and Transportation Times to Minimize Total Completion Time and Total Tardiness. *Expert Syst. Appl.* **2009**, *36*, 9625–9633. [CrossRef]

42. Tran, T.H.; Ng, K.M. A Hybrid Water Flow Algorithm for Multi-Objective Flexible Flow Shop Scheduling Problems. *Eng. Optim.* **2013**, *45*, 483–502. [CrossRef]
43. Allahverdi, A.; Aydilek, H.; Aydilek, A. No-Wait Flowshop Scheduling Problem with Two Criteria; Total Tardiness and Makespan. *Eur. J. Oper. Res.* **2018**, *269*, 590–601. [CrossRef]
44. Wan, L.; Mei, J.; Du, J. Two-Agent Scheduling of Unit Processing Time Jobs to Minimize Total Weighted Completion Time and Total Weighted Number of Tardy Jobs. *Eur. J. Oper. Res.* **2021**, *290*, 26–35. [CrossRef]
45. Allali, K.; Aqil, S.; Belabid, J. Distributed No-Wait Flow Shop Problem with Sequence Dependent Setup Time: Optimization of Makespan and Maximum Tardiness. *Simul. Model. Pract. Theory* **2022**, *116*, 102455. [CrossRef]
46. Aydilek, A.; Aydilek, H.; Allahverdi, A. Algorithms for Minimizing the Number of Tardy Jobs for Reducing Production Cost with Uncertain Processing Times. *Appl. Math. Model.* **2017**, *45*, 982–996. [CrossRef]
47. Najat, A.; Yuan, C.; Gursel, S.; Tao, Y. Minimizing the Number of Tardy Jobs on Identical Parallel Machines Subject to Periodic Maintenance. *Procedia Manuf.* **2019**, *38*, 1409–1416. [CrossRef]
48. Della Croce, F.; T'kindt, V.; Ploton, O. Parallel Machine Scheduling with Minimum Number of Tardy Jobs: Approximation and Exponential Algorithms. *Appl. Math. Comput.* **2021**, *397*, 125888. [CrossRef]
49. Hejl, L.; Šůcha, P.; Novák, A.; Hanzálek, Z. Minimizing the Weighted Number of Tardy Jobs on a Single Machine: Strongly Correlated Instances. *Eur. J. Oper. Res.* **2022**, *298*, 413–424. [CrossRef]
50. Behnamian, J.; Zandieh, M.; Fatemi Ghomi, S.M.T. Due Window Scheduling with Sequence-Dependent Setup on Parallel Machines Using Three Hybrid Metaheuristic Algorithms. *Int. J. Adv. Manuf. Technol.* **2009**, *44*, 795–808. [CrossRef]
51. Behnamian, J.; Fatemi Ghomi, S.M.T.; Zandieh, M. Development of a Hybrid Metaheuristic to Minimise Earliness and Tardiness in a Hybrid Flowshop with Sequence-Dependent Setup Times. *Int. J. Prod. Res.* **2010**, *48*, 1415–1438. [CrossRef]
52. Behnamian, J.; Zandieh, M. A Discrete Colonial Competitive Algorithm for Hybrid Flowshop Scheduling to Minimize Earliness and Quadratic Tardiness Penalties. *Expert Syst. Appl.* **2011**, *38*, 14490–14498. [CrossRef]
53. Otten, M.; Braaksma, A.; Boucherie, R.J. Minimizing Earliness/Tardiness Costs on Multiple Machines with an Application to Surgery Scheduling. *Oper. Res. Health Care* **2019**, *22*, 100194. [CrossRef]
54. Schaller, J.; Valente, J.M.S. Minimizing Total Earliness and Tardiness in a Nowait Flow Shop. *Int. J. Prod. Econ.* **2020**, *224*, 107542. [CrossRef]
55. Kellerer, H.; Rustogi, K.; Strusevich, V.A. A Fast FPTAS for Single Machine Scheduling Problem of Minimizing Total Weighted Earliness and Tardiness about a Large Common Due Date. *Omega* **2020**, *90*, 101992. [CrossRef]
56. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Kan, A.H.G.R. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. In *Annals of Discrete Mathematics*; Discrete Optimization II; Hammer, P.L., Johnson, E.L., Korte, B.H., Eds.; Elsevier: Amsterdam, The Netherlands, 1979; Volume 5, pp. 287–326.
57. Laborie, P.; Rogerie, J.; Shaw, P.; Vilím, P. IBM ILOG CP Optimizer for Scheduling: 20+ Years of Scheduling with Constraints at IBM/ILOG. *Constraints* **2018**, *23*, 210–250. [CrossRef]

applied sciences

Article

Assembly Line Optimization Using MTM Time Standard and Simulation Modeling—A Case Study

Matic Breznik [1], Borut Buchmeister [2] and Nataša Vujica Herzog [2,*]

[1] BSH Nazarje, Savinjska cesta 30, 3331 Nazarje, Slovenia; matic.breznik@student.um.si
[2] Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, 2000 Maribor, Slovenia; borut.buchmeister@um.si
* Correspondence: natasa.vujica@um.si; Tel.: +386-2-220-7635

Abstract: This study presents an approach to solving the assembly line balancing problem (ALBP) using the Methods-Time Measurement (MTM) time standard and simulation software. ALBP is a common problem in manufacturing where a set of tasks with fixed times must be assigned to a series of sequential workstations in order to minimize the total idle time and reduce the assembly cost per product. This study uses MTM, a widely used production process scheduling method, to create a new time analysis of an assembly process that was previously balanced using the Work-Factor method and time study. This literature review shows that there are a lack of combinations of updated time analyses with newer simulation approaches in the current literature, and this was the motivation for the present work. An assembly line simulation was performed using Simio software to evaluate different design options and operating scenarios. The results show that the use of MTM and simulation can help minimize idle time and improve assembly line performance, thereby reducing costs and increasing efficiency. This study shows that the approach of using MTM and simulation is effective in solving ALBP and is a useful tool for manufacturing companies to improve the performance of their assembly lines and reduce costs.

Keywords: optimization; production planning; assembly line; MTM time standard; simulation; industry 4.0

1. Introduction

An assembly line is a manufacturing process consisting of various tasks in which multiple parts are sequentially assembled into a product at several workstations to produce the final product. It is widely used in mass production for manufacturing various types of product, such as automobiles or electronic products. The workstations are arranged sequentially, with tasks being performed by workers simultaneously. The main layout problem is determining the optimal arrangement of tasks to workstations, which is commonly known as the assembly line balancing problem (ALBP). ALBP is a classical optimization problem in industrial engineering with the main objective of optimizing the efficiency (number of workstations and cycle time) of assigning tasks to workstations. This approach has been widely discussed and presented [1–5].

Setting up an assembly line is usually a long-term process that involves major investments. In addition to planning new assembly lines, existing assembly lines must also be periodically redesigned to accommodate various changes in the production process. Therefore, companies constantly face the challenge of changing their assembly systems quickly and economically.

In light of the ideas of Industry 5.0, with its focus on the human-friendly, human-centered, and sustainable design of production environments, our study focused on less-publicized approaches to time analysis of the assembly process. Although time analysis is a well-known approach, it has recently become more interesting because it is directly related to workers and their activities. Workers have different skills and perform tasks in different ways in terms of speed, motivation, and diligence, a topic that is also considered in ergonomics, and this could be a future direction of ALBP approaches. ALBP and time analyses are closely related, so new variations and combinations of the two approaches that are still being developed represent a new and interesting area for future research.

This study presents an approach to solving ALBP using the Methods-Time Measurement (MTM) time standard and simulation software. A new time analysis using MTM was performed to assess the current process and was balanced with the use of Work-Factor. The new time analysis was used to prepare the rebalancing of an assembly line. Scenarios before and after rebalancing were then compared in a simulated environment. The assembly line simulation was performed using Simio software to evaluate different design options and operating scenarios. The usefulness of the presented approach was demonstrated by running the assembly line in a three-shift serial production in a large manufacturing company for household appliances.

Although the focus of the present study is on time analysis, an in-depth analysis of the work and work methods, including an ergonomic evaluation using the European Assembly Worksheet (EAWS), was performed at the same time. Due to the complexity of the optimization process, which requires the consideration of many influencing factors, the present study was limited to time analysis.

2. Literature Review

2.1. Simulation Modeling

As a powerful tool for analyzing complex stochastic systems, of which assembly lines are undoubtedly one, computer simulation modeling has been widely used [6–9]. Simulation modeling is the process of creating and testing a computer-based mathematical model of a physical system. The main objectives of simulation modeling can be summarized as follows:

- Gaining insight into the operation of a system; it is difficult to study the system at standstill.
- Developing operational or resource strategies to improve system performance; existing operating systems should be improved.
- Testing new concepts and/or systems prior to implementation, and typically, testing how well the new proposed model will work and reviewing and refining the configuration of the selected equipment.
- Obtaining information without interfering with the actual system.

Simulation modeling has several advantages, but also some disadvantages that should be considered. The main advantage is the ability to examine systems dynamically and in real time during the simulation run, which usually takes less time than testing in a real environment. With a computer model, operation and interaction with different scenarios can be simulated in seconds. The reduced analysis effort allows practitioners to analyze many more different types of system than was previously possible. Simulation software with the ability to dynamically animate the operation of the model is useful for demonstrating system operation and troubleshooting potential failures. In addition to the obvious advantages, there are some disadvantages to simulation modeling.

Simulation cannot provide accurate results if the input data are inaccurate. Therefore, data collection is a very important part of the simulation process, even though it is often neglected. Another challenge is problem solving. Although the simulation model provides management with several possible solutions, simulation alone does not solve the problem—this is still the responsibility of management.

The use of various simulation software programs has been covered in several papers [10–13] and books [6,14,15]. One of the most important decisions in conducting a simulation study concerns the choice of software. We can choose a simulation package or a general-purpose programming language. The advantages of simulation packages can be summarized as follows:

- They provide all the segments needed to build a simulation model (shorter time and lower cost);
- The basic modeling constructs are usually more user-friendly (easy to use);
- Simulation models are easier to modify and maintain;
- Automatic checking for possible errors (better error detection).

In contrast, simulation models written in a general programming language have some other advantages:

- An analyst already knows the programming language;
- A model written in a specific programming language (C, C++, Java) can be better tailored to a specific application, and is therefore faster (shorter execution time);
- Greater flexibility in programming.

Discrete-event simulation packages are usually of two main types: general-purpose simulation packages and application-oriented simulation packages. The former can be used for any application, but may have specific features for certain applications (e.g., manufacturing or process optimization). The second type of simulation software is tailored to a specific type of application, such as manufacturing or healthcare.

2.2. Simulation Software

There are several general-purpose simulation packages on the market, such as Arena [16,17], ExtendSim [18], Simio [19,20], Anylogic [21,22], and others. Arena is typically used for applications such as manufacturing, supply chains, defense, and healthcare. The modeling structures are functionally organized into different "templates" that are used to model arrivals, departures, services, and unit decision logic. This is the basic approach. The Arena software also allows for an advanced process approach with access to external data files in Excel, Access, and SQL databases. A model is created by dragging modules into the main model window and connecting them together to represent the flow of units through the simulated system. Detailed modules can be designed using dialog boxes. Three-dimensional (3D) animations can be created simultaneously using the "Visual Designer". In addition, there is activity-based costing that provides value-added and non-value-added cost and time reports. The Arena simulation package has not been widely used in previous research, but there are two cases where it has been used for assembly lines [23,24].

ExtendSim is the name for a family of four general-purpose simulation packages targeted toward specific market segments. A simulation model is created in a virtual environment by selecting blocks from libraries, placing them in selected locations in the model window, similar to Arena, and connecting them together to represent the flow of units through the system. Detailed modules can be designed using dialog boxes. The software can create a wide range of different system configurations. It has activity-based costing, which allows fixed and variable costs to be assigned to a unit as it moves through the simulated system. The "Scenario Manager" allows us to explore different scenarios and shows us the model responses from one scenario to another. This literature review shows that there are some applications of ExtendSim simulation packages in logistics [25], transportation [26], and some other non-processing areas.

Simio is an object-oriented suite of simulation and planning products [20]. It is a simulation modeling framework based on smart objects that allow models to be created either using a standard library (for discrete-event simulations) or by allowing the user to create new objects. An object in a library can be a customer or a machine. A model is created by dragging objects into the 2D "Facility" window, connecting them with links to show the flow of entities through the simulated system. Similar to the previously presented software packages, detailed modules can be designed using a property editor. A 3D perspective view is also available for better visualization. The structure of an object in Simio is identical to the structure of a model, and each model is automatically a module that can be used to build hierarchical models. Specific properties (e.g., the machining time of a machine) and states (e.g., idle or busy) can be defined for each inserted object. Simio also provides a set of sophisticated functions for running and analyzing simulation experiments with different scenarios. The Simio simulation package is more widely used than the previous two simulation software packages, but not for assembly lines. It has some applications in logistics [27], service planning [28], production planning [29], and cellular manufacturing [30].

Anylogic is a powerful and flexible simulation software package that can be used to model and analyze complex systems in a variety of industries, including healthcare, transportation, and logistics. Anylogic offers users three different modeling approaches: discrete-event, agent-based, and system dynamics. This makes it a versatile tool that can be used to model a wide range of systems. Like other software solutions, Anylogic also has powerful visualization and analysis tools, including the ability to create 2D and 3D animations of models, and perform advanced statistical analysis. There are some previous research papers that use the Anylogic simulation package, but it is used for manufacturing simulation, lean six sigma implementation, value stream analysis, lean manufacturing, smart factories, material handling, and cloud simulation [31–35].

2.3. Assembly Line Balancing

The assembly line balancing problem (ALBP) deals with the allocation of tasks to workstations while optimizing one or more previously selected goals, without violating precedence constraints on the tasks or other constraints imposed on the assembly line.

The simple assembly line balancing problem (SALBP), which envisions a single-mode line with a fixed cycle time and deterministic task times, does not consider all the features typical of real problems. Real AL problems should consider several additional features, such as cost functions, equipment selection, parallelization, stochastic task times, and others, and can be solved via the so-called generalized assembly line balancing problem (GALBP).

The mixed-model assembly line balancing problem (MALBP) can be considered a special case of GALBP where several similar models, which are variations of the same basic product and differ only in certain adaptable product properties, can be assembled simultaneously and continuously [36].

Parallelization is another feature that should be considered in assembly line configurations. Different forms of parallelization can occur in the real environment and increase the solution space of the problem, such as parallel lines [37], parallel workstations performing the same set of tasks [38], or parallel bipartite lines [39]. Parallel assembly sequence planning (PASP) is best explained in the work of Gulivindala [40,41]. PASP is treated as an NP-hard problem because of methodological difficulties in its development phase and computational complexity in its implementation for solution generation.

Based on the assembly line layout, two types of assembly line balancing are known: the one-sided assembly line balancing problem and the two-sided assembly line balancing problem (TALBP), the latter of which is widely used in the assembly of large products such as busses and trucks [42]. TALBP is characterized by a set of tasks that must be divided and processed at a series of paired stations, where each station contains two opposing workstations with two workers [43]. The TALBP approach, with four specific constraints (precedence, cycle time, allocation, and direction), is much more complex than the one-sided ALBP approach, although simple ALBP with minimization of the number of workstations is already an NP-hard problem [44]. The methods used to solve ALBP can be divided into three groups: exact methods, heuristic methods, and meta-heuristic methods. With all the different possibilities and approaches, the field of ALBP solutions continues to grow and evolve [45].

NP-hard problems are described as a class of computational problem for which a solution can be verified in polynomial time, but for which no known algorithm can solve the problem in polynomial time. NP-problems are of great interest in computer science because many important problems, such as the traveling salesman problem and the satisfiability problem, are known to be NP-problems. The class NP is defined as a set of decision problems for which a proposed solution can be verified in polynomial time. This means that the amount of time required to solve or verify the problem grows faster than a polynomial function of the size of the input [46].

2.4. Assembly Line and Industry 4.0

In addition to classical mathematical approaches, new technologies in the context of Industry 4.0 (I4.0) can accelerate workplace design on the assembly line. Industry 4.0 is leading companies to gradually and continuously automate traditional manufacturing processes [12], but at the same time, Industry 5.0, with its focus on a more human-centered approach to industrial work system design, is just around the corner [47]. Industry 5.0 emphasizes explicit attention to consequences for employees in the system in order to design production environments that are human-friendly, human-centered, and sustainable.

Assembly systems are also affected by this revolution, with the concept of Assembly System 4.0 (AS4.0) aiming to improve performance and workplace design [9,48–55].

Traditional approaches to optimizing assembly systems mainly consider time and cost variables, but some studies also consider ergonomic aspects [56]. On the assembly line, the work is repetitive and requires the full attention of the worker. The processes are flexible and must also be organized while taking into account the workers' skills [57]. Workers are an integral and very important part of production systems, as they still perform the majority of operations. They usually have different skills and expertise and perform tasks in different ways in terms of speed, motivation and diligence.

The current trend of Industry 4.0, with "smart" paradigms such as sensors, communication platforms, simulation, data-intensive modeling, and predictive engineering, offers us the opportunity to recreate the work environment in a virtual scenario where it is possible to simulate manual activities, evaluate ergonomic metrics, and perform time analyses simultaneously [58]. Software-based time analyses and simulations allow us to study different operating scenarios before setting up an assembly line in a real environment, and also represent the added value of new technologies.

2.5. Assembly Line Reliability

System reliability should also be considered in the optimal design of assembly lines. Optimizing system reliability involves formalizing and continuously improving the methods and techniques used to address the reliability of a complex system [1]. Within assembly lines, system reliability refers to cycle time uncertainty. Changes in cycle time (e.g., increases in cycle time) can cause a line to become unbalanced, resulting in production losses. Therefore, maximizing assembly line reliability is another important goal for ALBP, in addition to minimizing cycle time.

3. Problem Definition

The latest product produced on an observed assembly line, presented in Figure 1, exceeded the expected production quantity. This was the result of the unforeseen positive diffusion of a new technology on the market. Since the new planned output could not be achieved on the current assembly line, the question arose as to whether the current task workflow from one workstation to another was optimally balanced or whether it was necessary to order a twin of the current assembly line.

The product assembled on the production line (Figure 1) is a fully automated espresso machine with external dimensions of $400 \times 300 \times 400$ (cm $\times$ cm $\times$ cm). There are approximately 216 operations composed of 864 steps of time analysis, which need to be completed to assemble the whole product. The process time for assembly is 1102 s.

This problem is commonly known as the assembly line balancing problem and is best described in [7]. The ALBP is represented by a set of tasks with fixed times that must be assigned to a set of sequential workstations. The order in which the tasks can be executed is constrained by a set of priority relationships. The most important constraint to consider is the time available at each station, referred to as the cycle time.

Figure 1. The latest product assembled on a production line was a fully automated espresso machine.

If the station utilization is lower than the cycle time, then the idle time for each cycle is the difference between the station utilization and the assembly line cycle time. The goal of ALBP's solution is to minimize the total idle time across all stations on the assembly line. In this way, the assembly cost per product is reduced to a minimum [59,60].

For better knowledge of all project participants, a new assembly time analysis was created for this product. It was based on the Methods-Time Measurement time standard. The newly created time analysis was used to compare the current state with the improvement possibilities on the assembly line. Similar to [61], MTM was used because of its use in production process planning. It assumes that individual basic movements in the correct sequence are more time-efficient and can be performed with fewer errors.

After rebalancing of the assembly line, the European Assembly Worksheet (EAWS) was used to assess workers' posture and movements [62,63]. EAWS is a screening tool used to assess workers' biomechanical workload (postures, forces, manual handling, and repetitions) that identifies key ergonomic issues and provides the opportunity to develop solutions to overcome them. Although ergonomic assessment was a part of our work analysis, the ergonomic results are not presented in detail in this paper.

The assembly line was split into three logical zones: A, B, and C. To obtain the results of the study as soon as possible and implement the improvements, only zones A and B were included in the study, because the cycle time of zone C was lower than the new planned cycle time of the assembly line. The layout zones are highlighted in Figure 2.

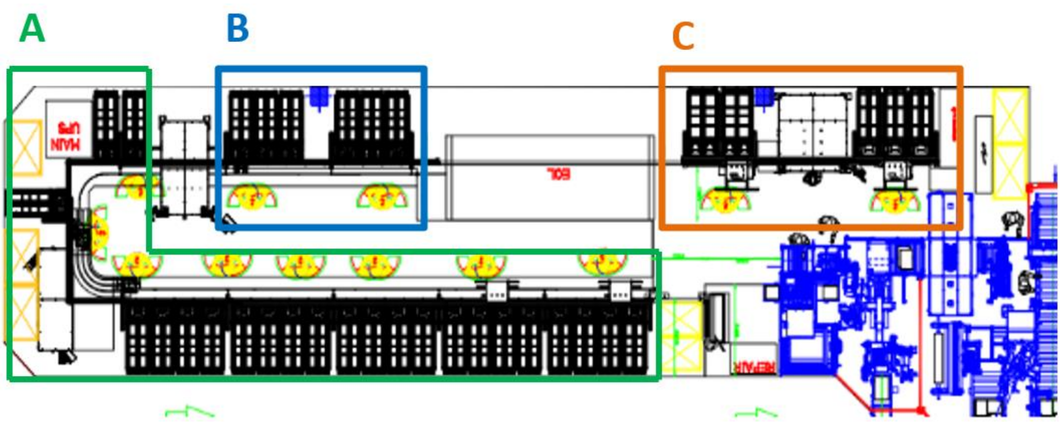


Figure 2. Layout of the assembly line, with the assembly zones highlighted.

The assembly line was divided into three logical zones based on the effect of the fully automated inspection stations used on the assembly line. Zone A included eight workstations. After Zone A, an automated PreCheck camera station was used to perform a visual inspection of all assembly work performed to date. This was the last step before sealing the product with the housings. Zone B included 2 workstations where assembly of the housings and preparation of the product took place prior to the fully automated functional and safety testing of the product. Zone C represents the last two workstations where the finishing operations of assembly and packaging were performed, which are not relevant to the possible safety-critical aspects of the product.

To minimize disruptions on the assembly line, which was running in a three-shift serial production, it was decided to display all collected information in a simulated environment. The simulated environment was prepared using Simio simulation software.

The main goal of the study was to simulate whether the new desired output of the assembly line can be achieved, without disturbing the current assembly line, which is in serial production.

Study Limitations and Assumptions

Our study was limited to the production line in a selected medium-sized company, so that all specifics and characteristics were limited to the type of production and equipment. The other limitations of the present study are as follows:

- We came to an agreement with the manufacturer that prevents the publication of some data.
- The precedence graph for the assembly had to be respected.
- Because of the test stations, the operations had to remain in their designated zones. This means that operations that were in Zone A before rebalancing remained in Zone A after rebalancing.
- Two of the operations in the assembly line required oiling and had fixed oiling devices. To reduce costs and ease the transition to the new assembly order, these operations remained at the same workstations.

This study's assumptions are based, first, on the anticipated need to increase assembly line output due to increased market demand. Second, the MTM time analysis is assumed to accurately reflect the time required for each step of the assembly operations.

4. Materials and Methods

The ALBP was analyzed using the MTM time standard. This is a widely used method for scheduling production processes because it assumes that individual basic movements in the correct sequence are more time-efficient and can be executed with fewer errors. Due to the cycle times in the observed assembly line, the use of the MTM Universal Analysis System (MTM-UAS) was recommended. This method of predetermined motion time system was described in the last review article [64] as state-of-the-art in practice.

The Work-Factor (WF) system and the Methods-Time Measurement (MTM) system are well known approaches from a group of predetermined times. Both systems have some benefits and obstacles and have undergone many changes and variants over time. The MTM system was first described by Maynard, Stegemerten, and Schwab [65] for production process planning. Both approaches, WF and MTM, are based on the assumption that individual basic movements can be performed in the correct sequence more time efficiently and with fewer errors. The time units used are referred to as TMUs (Time Management Units). The components of the process follow each other in a linear sequence, which is reflected in their sequential process.

The basic concept of MTM is to decompose a task into its basic human activities, use basic times for them from tables, and combine them into a basic time for the entire task. Several variants of MTM have been developed, differing in their focus. Because of the length of the process time, the MTM Universal Analysis System (MTM-UAS) was chosen in this study (Figure 3). MTM-UAS, which assigns a pre-determined time standard to specific individual process steps, was created to meet manufacturers' demands for productivity improvement in batch production. Today, it is widely used in the automotive industry [66]. The categorization MTM-UAS aims for serial production and minimization of the time units used. Sequences of movements and tasks occur regularly and in rapid succession. Taylor and Gilbreth analyzed the ideal workflow from the point of view of efficiency. In addition to the personal skills and abilities of the workers, they emphasized the sequence of work steps as particularly important [67].

Data cards — MTM-UAS — Basic operations

Get and Place			DC	1	2	3
			Code	TMU		
≤ 1 kg	easy	approx.	AA	20	35	50
		loose	AB	30	45	60
		tight	AC	40	55	70
	difficult	approx.	AD	20	45	60
		loose	AE	30	55	70
		tight	AF	40	65	80
	handful	approx.	AG	40	65	80
> 1 to ≤ 8 kg		approx.	AH	25	45	55
		loose	AJ	40	65	75
		tight	AK	50	75	85
> 8 to ≤ 22 kg		approx.	AL	80	105	115
		loose	AM	95	120	130
		tight	AN	120	145	160
Place			Code	1	2	3
		approx.	PA	10	20	25
		loose	PB	20	30	35
		tight	PC	30	40	45

Handle Tool	Code	1	2	3
approximate	HA	25	45	65
loose	HB	40	60	75
tight	HC	50	70	85
Operate	Code	1	2	3
simple	BA	10	25	40
compound	BB	30	45	60
Motion Cycles	Code	1	2	3
one motion	ZA	5	15	20
motion sequence	ZB	10	30	40
re-position and one motion	ZC	30	45	55
tighten or loosen	ZD	20		
Body Motions	Code			
Walk / m	KA	25		
Bend, stoop, kneel (incl. arise)	KB	60		
Sit and Stand	KC	110		
Visual Control	VA	15		

Figure 3. Part of code selection form according to the MTM-UAS procedure.

The cycle times of the observed production line range from 60 to 120 s, and the detailed MTM analysis with all the required steps was extensive. To illustrate the application of MTM analysis, a brief example based on a laboratory demonstration of a workstation was prepared and is shown in Figure 4. The example shows 7 assembly operations consisting of 14 assembly steps. For each movement that is performed, the starting position of the worker must be known. Code is then developed based on factors such as the difficulty of grasping and positioning the object, which depend on the size, weight, and type of packaging of the object. Accurately measuring the length of the movement is critical to determining the specific code for each step. Additionally, codes for body motion, including side steps, are used in this example.

MTM-UAS allows for the creation of building blocks that represent operations that are repeated in the assembly process. It is recommended that these building blocks are used to avoid discrepancies in the different phases of the timing analysis.

Originally, the analysis was divided into seven parts. Splitting the timing analysis is recommended when the results show that the output is not achieved and realignment of the assembly line is required. The split parts were later combined into one workstation for presentation purposes [58]. The sum of all steps in the example in Figure 4 results in a base time of 21.96 s. The base time represents the cycle time of the workstation.

In our analysis, the European Assembly Worksheet (EAWS) was used to assess workers' posture and movements. EAWS was originally developed for the assessment of assembly work in the automotive industry, where work is performed in short cyclic segments. The EAWS structure consists of the following sections: the general section, the working posture section, action forces, manual material handling, and repetitive upper limb movements. The EAWS index is composed of two values: one related to the whole body and the other to the upper limbs. The risk zones are classified as follows:

MTM Operation Sheet	**Product BOP**	ES69		**B/S/H/**	
	Structure	Example			
	Owning User	Breznik, Matic [FCPN-OSBI]			
Basic Time (sec)	21.96				
No	Code	Description	Time	Q*F	Total [sec]
1	AB2	Assemble Service door	45	1 x 1.0	1.62
2	PC1	Position Service door	30	1 x 1.0	1.08
3	AC2	Assemble Water tank	55	1 x 1.0	1.98
4	AB2	Assemble Brita filter	45	1 x 1.0	1.62
5	KA	Side step	25	1 x 1.0	0.9
6	AB2	Assemble Brita filter aid	45	1 x 1.0	1.62
7	AF3	Assemble Water tank cover	80	1 x 1.0	2.88
8	PC1	Position Water tank cover	30	2 x 1.0	2.16
9	AF3	Assemble Bean tank cover	80	1 x 1.0	2.88
10	PC1	Position bean tank cover	30	1 x 1.0	1.08
11	KA	Side step	25	1 x 1.0	0.9
12	AB2	Assemble Drip pan	45	1 x 1.0	1.62
13	PA2	Position Drip pan	20	1 x 1.0	0.72
14	KA	Side step	25	1 x 1.0	0.9

Figure 4. Example of MTM time analysis of a workstation.

- Green, 0–25 points, low risk—no action is needed,
- Yellow, 26–50 points, moderate risk—further risk assessment and analysis are performed, taking into account additional risk factors (redesign or recovery actions)
- Red, >50 points, high risk—action to reduce the risk is required.

For our simulation, final solutions with a total score below 25 were considered (Figure 5).

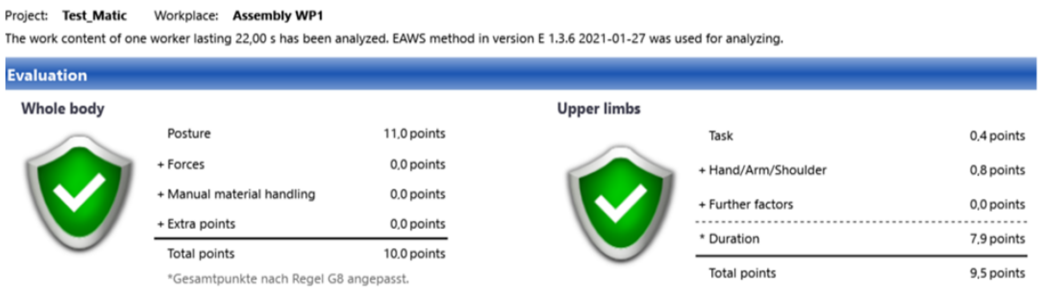


Figure 5. Example of EAWS ergonomic analysis of a workstation.

To evaluate the accuracy of the prepared time analysis, a statistical analysis of a previous time analysis project was performed. The collected data of the project, which is now in serial production, were used to create a time distribution using ExpertFit software.

Using ExpertFit, any analyst, regardless of prior knowledge of statistics, can avoid pitfalls that undermine the success of simulation studies. ExpertFit identifies the best of the probability distributions under consideration and helps the analyst decide whether the fit is a good one. If no adequate fits are found, ExpertFit [68] can be used to create an empirical distribution function.

To minimize disruptions on the assembly line during three-shift serial production, new scenarios created using the MTM method were tested in a simulated environment. Simio simulation software was used to prepare the simulated environment. One of the main advantages of Simio is that it allows users to test and evaluate different design options and operating scenarios to determine the best solutions to improve performance and reduce costs [19].

The assembly line in question met most of the conditions that determine whether it is appropriate to build a simulation model. The reasons for this were the longer simulated time interval of the assembly line process under study and the variant planning without interfering with the real system [69].

As mentioned earlier, a model of the assembly line was created using Simio simulation software. The assembly line model was then used to simulate different scenarios (Figure 6). The effects of variables on the performance of the assembly line and its bottleneck stations were simulated. The simulation results had to be critically analyzed, and the lessons learned had to be applied to the real environment in which the assembly line was optimized.

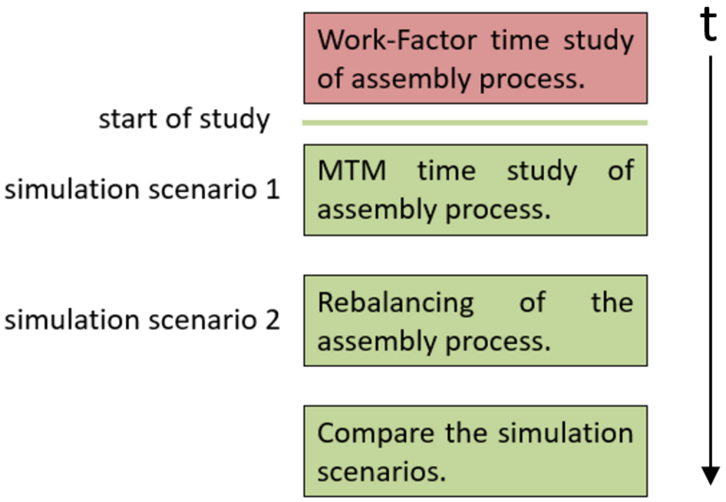


Figure 6. Explanation of the two simulation scenarios that were later used in Simio.

Simio simulation software was used because we are familiar with it and have had good experience using it to study various problems in manufacturing. Similar to [27], it was chosen for its ability to represent the system in three dimensions and model realistic spatial relationships in the layout. This resulted in a user-friendly interface that facilitated model verification and validation.

5. Results

5.1. Line Balancing

The previous time analysis was performed by another engineer using the Work-Factor method. To keep the rebalancing results as accurate as possible, a new time analysis was created from scratch using MTM.

The cycle times of the observed production line were between 60 and 120 s. Due to the nature of the assembly and the length of the cycle times at the workstation, the use of the MTM Universal Analysis System (MTM-UAS) was recommended. The assembly operations were simple and were all performed manually, except for the screw driving operations, which were performed using electric screwdrivers. No above-average training or certification was required to perform nine of the ten jobs mentioned. The one that required more in-depth training took place at Workstation 9, which was responsible for handling the results and responding appropriately to the results of the automated PreCheck station.

The time analysis was prepared by an expert engineer, but there was still a possibility that the results prepared by different engineers may differ. To evaluate the accuracy of the engineers' time analyses, a statistical analysis of a previous time analysis project was performed.

The cycle time for a particular workstation was estimated to be 59.8 s. After the new work procedure was implemented and the learning curve of the two weeks was completed, 14 cycles were measured, which are shown in Table 1.

Table 1. Cycle times measured on a different project for which the same expert engineer had prepared the assembly line balancing with the use of MTM.

Cycle (#)	Cycle Time (s)	Cycle (#)	Cycle Time (s)
1	72	8	49
2	62	9	55
3	54	10	58
4	69	11	55
5	62	12	60
6	66	13	64
7	53	14	58

Based on the times recorded and shown in Table 1, a time distribution was generated using ExpertFit software. The results of the distribution show that the normal distribution 59.643; 6.105; delta 0.401 is the best fit for the acquired cycle times.

After comparing the engineer's estimated time based on method MTM-UAS and the distribution of times measured on the assembly line, there is a 0.26% difference in the results. The results show that the method and the expert engineer are suitable for preparing the time analysis for this type of assembly production. The results of the MTM-UAS method, which was prepared for the assembly line in question, were used in the simulation model.

After the time analysis was prepared, the assembly line adjustment had to be planned, also known as the rebalancing process. Each task and each step of the tasks were coded using MTM-UAS and organized using the EasyPlan software. Examples of the coding, determination of frequency, determination of quantity, and time of the steps (TG) are shown in Table 2. The TG Sum column shows the assigned time of operations in seconds.

Table 2. Example of an operation divided into four steps, analyzed using the MTM-UAS code, and organized using the EasyPlan software.

Step	Code	Q × F	TG (s)	TG Sum (s)
1	KA	1 × 2	0.90	1.80
2	AC2	1 × 1	1.98	1.98
3	AB2	1 × 1	1.62	1.62
4	PC1	1 × 2	1.08	2.16

When tasks were assigned to a workstation, the sum of all steps of the tasks at each workstation must be less than the desired cycle time of the assembly line.

The results of the new time analysis were combined with the original assembly process and are shown in Table 3.

Table 3. Cycle times by workstation—before rebalancing.

Workstation (#)	Cycle Time (s)	Workstation (#)	Cycle Time (s)
1	107.1	6	86.32
2	95.04	7	77.40
3	96.48	8	86.58
4	99.74	9	95.42
5	73.80	10	95.58

After seeing the results shown in Table 3, rebalancing was performed. The results of the rebalancing are shown in Table 4.

The bottleneck for the new rebalancing in Zone A is workstation 1, which is the pacemaker of the assembly line. The problem is that Zone B, with workstations 9 and 10, is the real bottleneck of the assembly line. For further improvements and studies on the assembly line, a worker would need to be added to Zone B. This would mean that a lot of workers would have to be added in Zone A to maintain the desired workstation occupancy.

Table 4. Cycle times by workstation—after rebalancing.

Workstation (#)	Cycle Time (s)	Workstation (#)	Cycle Time (s)
1	93.96	6	86.32
2	92.52	7	86.58
3	92.16	8	86.58
4	90.20	9	95.42
5	93.78	10	95.58

5.2. Simulation Model

The purpose of creating a simulation model of the assembly line was to study the effects of rebalancing on the efficiency of the assembly process. To improve the quality of the results obtained using time analysis, a simulation model was created using the time analysis data. The additional benefit of the simulation model is its consideration of the influences and disturbances that are present in an assembly environment. In the simulation model, they are represented by:

- The calculated distribution of the time analysis data;
- The first pass yield of the workers and the inspection stations in the first run;
- The repair station effect;
- The work schedule;
- Breaks.

5.2.1. Preparation of the Simulation Model

A simulation model of the assembly line was created using Simio simulation software. The model contained all the information that could be collected in the real environment.

Figure 7 shows the layout of workstations 1–10 represented in the rebalancing data. The layout also includes workstations 11 and 12, which were excluded from the analysis due to their clear lower cycle time, and which are repair stations to which the devices must go if they fail the device tests at the PreCheck, End of Line, or Complete stations.

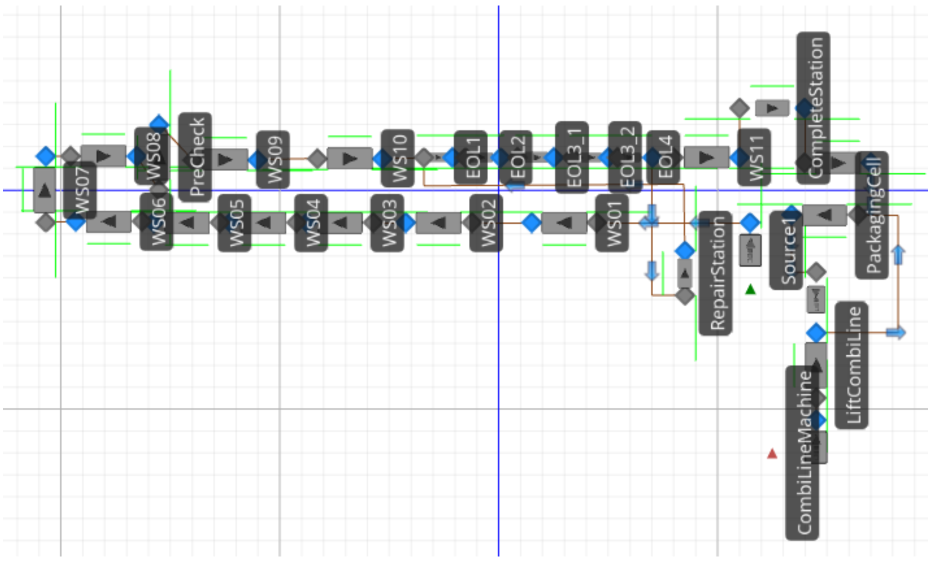


Figure 7. Layout of the assembly line presented in Simio simulation software.

The first pass yields (FPY) of each test station and the average repair times were also used in the simulation models.

Because the calculated distribution of the previously prepared time analysis reflects the repeatability of the prepared time analysis, the calculated normal distribution was included in the simulation model for each workstation cycle time.

Two sources were used because the packaging cell was shared by two assembly lines to check if there were problems with the availability of the packaging cell.

A three-shift schedule was implemented in the simulation model, which is shown in Table 5. The time spent on breaks, lunch breaks, cleaning the workplace, and talking to the production manager affects the final production and must be included in the simulation model.

Table 5. Work schedule followed at the assembly line.

Start Time	Duration (min)	End Time
06:11	79	07:30
07:40	110	09:30
09:55	115	11:50
12:00	110	13:50
14:11	79	15:30
15:40	110	17:30
17:55	115	19:50
20:00	110	21:50
22:11	79	23:30
23:40	110	01:30
01:55	115	03:50
04:00	110	05:50

To test the effects of rebalancing, two simulation scenarios were created. The first scenario used the assembly process before rebalancing, and the second scenario used the assembly process after rebalancing.

Figure 8 presents the 3D view in the simulation software Simio. Workstations in idle states are presented in gray, and the bottleneck effect is presented in yellow color as the buffer becomes filled; the entities that represent the product flow through the assembly workstations can be tracked.

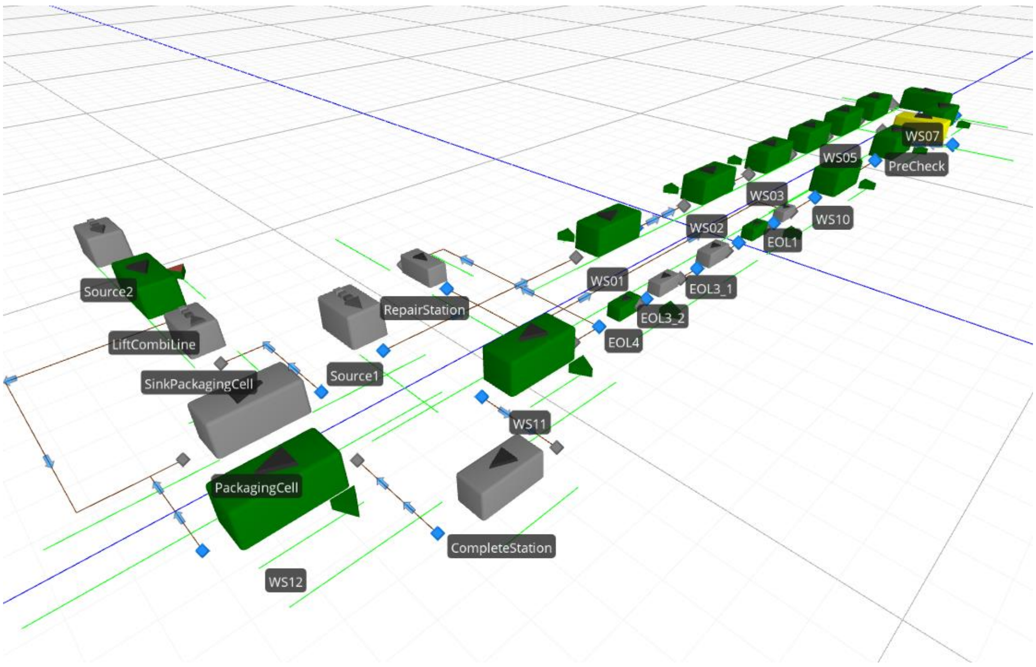


Figure 8. Three-dimensional view of the assembly line during running of a simulation scenario.

5.2.2. Results of the Simulation Scenarios

The PreCheck station split the assembly process into two logical parts, since the tasks of workstations 9 and 10 could not be executed before the tests of correct assembly had been performed on the device in the PreCheck station. For this reason, the results of the simulation scenario and the distribution of balance on the assembly line were divided into two separate parts, as shown in Tables 6 and 7.

Table 6. Dispersion of assembly line balance before rebalancing.

	Workstations 1–8	Workstations 9–10
Minimal occupancy (%)	67.92	86.11
Maximal occupancy (%)	100	86.46
Dispersion (%)	47.23	0.41

Table 7. Dispersion of assembly line balance after rebalancing.

	Workstations 1–8	Workstations 9–10
Minimal occupancy (%)	88.58	96.10
Maximal occupancy (%)	100	96.53
Dispersion (%)	12.89	0.45

After running the simulation scenario for a simulated month of assembly, average output rates per shift were created. The assembly line output per shift increased from 218 to 243 products per shift after rebalancing.

6. Discussion

This study aimed to evaluate the performance of an assembly line by analyzing the current task workflow and rebalancing it, if necessary, to achieve optimal production quantity. The study utilized the MTM-UAS time standard and simulation software Simio to perform a new time analysis and assess the effectiveness of the rebalancing process. The results show that the current assembly line was not optimally balanced and required rebalancing to improve production efficiency.

The distribution of the results of the time analysis, prepared by an expert engineer, shows that the decision to use MTM-UAS was correct and best suited to the type of assembly line that was observed.

The results of the new time analysis show that there was a clear need for rebalancing, as there was a 38.37% difference between the bottleneck workstation and the least occupied workstation. After rebalancing, the difference in occupancy between the new bottleneck workstation and the new least occupied workstations was reduced to 8.52%.

To confirm the idea of rebalancing, a simulation model was created to have as little negative impact as possible on the actual assembly line before we could make sure that the rebalancing decision was correct.

There is a slight difference between the previous results of the timing analysis due to all the other variables that the simulation model can also include. The results of the simulation scenario before rebalancing showed that there was a 47.23% difference in occupancy between the bottleneck workstation and the least occupied workstation. The results of the simulation scenario after rebalancing showed that the difference in occupancy between these workstations was reduced to 12.89%.

The simulation scenarios also had an impact on the performance of the assembly line, which was increased by 11.4%. This means that a twin assembly line is still needed, but a reduction from the planned three shifts to two on the twin assembly line can be realized.

All this was achieved without any disturbances to the observed assembly line that was in serial production, which was the main goal of the study.

Changes have already been made to test the impact of the new balance on the current assembly line. Work steps were changed between workstations so that the learning curve was rerun. The planned output from the simulation scenario results was achieved, and there was positive feedback from the assembly workers as the differences in workstation occupancy were reduced and clearly visible from their point of view.

This study showed the optimization of balancing results checked using simulation scenarios, but ALBP was still solved by hand. The next step is to use genetic algorithms to solve ALBPs that outperform existing heuristics [70]. In addition, the ergonomic design of the assembly line is an important component of a stable production process. Therefore, the ergonomic weighting of tasks should also be included in the process of solving an ALBP.

7. Conclusions

Under the conditions of never-ending competition, the importance of an efficient production system has become more and more important. The need for reductions in production costs and quick responses to customer demand have forced us to look for new approaches and solutions. Assembly lines, with their specifics of design, balancing, and scheduling for mass production, are an important part of these optimization efforts. The case presented, using an existing simulation software package, shows us a useful approach to successful assembly line optimization.

Production system simulation modeling is a powerful tool for optimizing manufacturing processes and improving productivity. It involves creating a mathematical model of the production system and simulating its behavior to study the effects of various changes on the system. For more detailed and tailored solutions, the use of metaheuristic methods should also be considered. Metaheuristic methods are powerful optimization techniques that have become increasingly popular in recent years. These methods are often used to solve complex optimization problems that cannot be solved using traditional optimization techniques. Traditional optimization techniques are often better suited to well-structured problems, while metaheuristics are better suited to complex, unstructured problems. In addition, traditional techniques are often deterministic and provide theoretical guarantees on the quality of the solution, while metaheuristics are often stochastic and do not provide such guarantees.

The results of this study could be useful to both academia and industry. For researchers, our study provides insight into the effectiveness of various methods for balancing assembly lines, and can serve as a basis for further research. For practitioners, our work can serve as a basis for decision-making processes and help determine the most appropriate approach to balancing assembly lines for their specific needs. In addition, the methodology we used can be applied to other similar systems and help improve their efficiency. Overall, we hope that our study contributes to ongoing efforts in the field of industrial engineering and stimulates further research and practical applications.

For future research, the latest industrialization concept, known as Industry 5.0, should also be considered in the development of industrial manufacturing. Its focus is on integrating advanced technologies with human-centered values to enable smarter, more efficient, and more sustainable production processes. Ergonomics should also be considered as an appropriate tool for worker assignment, taking into account their experience and physical abilities. Industry 5.0, characterized by the integration of cyber–physical systems, artificial intelligence, machine learning, and Big Data analytics with human operators, will be the focus of further research to create a more collaborative and personalized production environment. The results of this new research could help improve worker safety and comfort, reduce the risk of musculoskeletal disorders, and increase overall productivity.

Author Contributions: Conceptualization, N.V.H., B.B. and M.B.; methodology, M.B. and B.B.; software, M.B.; validation, N.V.H. and M.B.; formal analysis, M.B.; investigation, M.B.; resources, M.B.; data curation, M.B.; writing—original draft preparation, N.V.H. and M.B.; writing—review and editing, N.V.H., B.B. and M.B.; visualization, M.B.; supervision, N.V.H.; project administration, N.V.H.; funding acquisition, N.V.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Slovenian Research Agency within the framework of Grant P2-0190.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Y.; Peng, R.; Kucukkoc, I.; Tang, X.; Wei, F. System reliability optimization for an assembly line under uncertain random environment. *Comput. Ind. Eng.* **2020**, *146*, 106540. [CrossRef]
2. Zhang, Z.; Tang, Q.; Ruiz, R.; Zhang, L. Ergonomic risk and cycle time minimization for the U-shaped worker assignment assembly line balancing problem: A multi-objective approach. *Comput. Oper. Res.* **2020**, *118*, 104905. [CrossRef]
3. Battaïa, O.; Dolgui, A. A taxonomy of line balancing problems and their solutionapproaches. *Int. J. Prod. Econ.* **2013**, *142*, 259–277. [CrossRef]
4. Sivasankaran, P.; Shahabudeen, P. Literature review of assembly line balancing problems. *Int. J. Adv. Manuf. Technol.* **2014**, *73*, 1665–1694. [CrossRef]
5. Bryton, B. *Balancing of a Continuous Production Line*; Northwestern University: Evanston, IL, USA, 1954.
6. Law, A.M. *Simulation Modeling and Analysis*, 5th ed.; McGraw-Hill Education: New York, NY, USA, 2015.
7. Hoffmann, T.R. Assembly line balancing: A set of challenging problems. *Int. J. Prod. Res.* **1990**, *28*, 1807–1815. [CrossRef]
8. Ojstersek, R.; Javernik, A.; Buchmeister, B. Importance of Sustainable Collaborative Workplaces—Simulation Modelling Approach. *Int. J. Simul. Model.* **2022**, *21*, 627–638. [CrossRef]
9. Tanasic, Z.; Janjić, G.; Sokovic, M.; Kušar, J. Implementation of the Lean Concept and Simulations in SMEs—A Case Study. *Int. J. Simul. Model.* **2022**, *21*, 77–88. [CrossRef]
10. Negahban, A.; Smith, J.S. Simulation for manufacturing system design and operation: Literature review and analysis. *J. Manuf. Syst.* **2014**, *33*, 241–261. [CrossRef]
11. Robinson, S. Conceptual Modeling for Simulation: Issues and Research Requirements. In Proceedings of the 2006 Winter Simulation Conference, Monterey, CA, USA, 3–6 December 2006; pp. 792–800. [CrossRef]
12. Fischbein, S.; Yellig, E. *Why Is It So Hard to Build and Validate Discrete Event Simulation Models of Manufacturing Facilities?* Springer: New York, NY, USA, 2011; pp. 271–288. [CrossRef]
13. Baines, T.; Mason, S.; Siebers, P.-O.; Ladbrook, J. Humans: The missing link in manufacturing simulation? *Simul. Model. Pract. Theory* **2004**, *12*, 515–526. [CrossRef]
14. Thomopoulos, N. *Assembly Line Planning and Control*; Springer: Berlin/Heidelberg, Germany, 2014. [CrossRef]
15. Chung, C.A. *Simulation Modeling Handbook: A Practical Approach*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2003. [CrossRef]
16. Kelton, W.D.; Sadowski, R.P.; Sturrock, D.T. *Simulation with Arena*, 4th ed.; McGraw-Hill: Boston, MA, USA, 2007; 630p.
17. Rockwell Automation. Arena Simulation Software. Available online: https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html (accessed on 15 May 2023).
18. Imagine That, Inc. ExtendSim Simulation Software. Available online: https://extendsim.com/ (accessed on 15 May 2023).
19. Smith, J.S.; Sturrock, D.T.; Kelton, W.D. *Simio and Simulation: Modeling, Analysis, Applications*; Simio LLC: Sewickley, PA, USA, 2018.
20. Simio LLC. Simio Simulation Software. Available online: https://www.simio.com/ (accessed on 15 May 2023).
21. Daniyan, I.; Mpofu, K.; Ramatsetse, B.; Zeferino, E.; Monzambe, G.; Sekano, E. Design and simulation of a flexible manufacturing system for manufacturing operations of railcar subassemblies. *Procedia Manuf.* **2021**, *54*, 112–117. [CrossRef]
22. Antonova, V.; Grechishkina, N.; Kuznetsov, N. Analysis of the Modeling Results for Passenger Traffic at an Underground Station Using AnyLogic. *J. Commun. Technol. Electron.* **2020**, *65*, 712–715. [CrossRef]
23. Bongomin, O.; Mwasiagi Igadwa, J.; Oyondi Nganyi, E.; Nibikora, I. Optimal design of garment assembly line using simulation-based optimization. *Mendeley Data* **2019**, *10*, T5W96KH5W7. [CrossRef]
24. Cortés, P.; Onieva, L.; Guadix, J. Optimising and simulating the assembly line balancing problem in a motorcycle manufacturing company: A case study. *Int. J. Prod. Res.* **2010**, *48*, 3637–3656. [CrossRef]
25. Straka, M.; Khouri, S.; Lenort, R.; Besta, R. Improvement of logistics in manufacturing system by the use of simulation modelling: A real industrial case study. *Adv. Prod. Eng. Manag.* **2020**, *15*, 18–30. [CrossRef]
26. Bindzar, P.; Macuga, D.; Brodny, J.; Tutak, M.; Malindzakova, M. Use of Universal Simulation Software Tools for Optimization of Signal Plans at Urban Intersections. *Sustainability* **2022**, *14*, 2079. [CrossRef]
27. Achkar, V.G.; Bär, V.; Cornú, F.; Méndez, C.A. A Simulation-Based Tool to Support Decision-Making in Logistics Design of a Can Packaging Line. *Int. J. Food Eng.* **2020**, *16*, 20170089. [CrossRef]
28. Longfei, Z.; Zhang, L.; Ren, L.; Wang, J. Real-Time Scheduling of Cloud Manufacturing Services Based on Dynamic Data-Driven Simulation. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5042–5051. [CrossRef]
29. Ojstersek, R.; Tang, M.; Buchmeister, B. Due date optimization in multi-objective scheduling of flexible job shop production. *Adv. Prod. Eng. Manag.* **2020**, *15*, 481–492. [CrossRef]
30. Sadeghi, A.; Suer, G.; Sinaki, R.Y.; Wilson, D. Cellular manufacturing design and replenishment strategy in a capacitated supply chain system: A simulation-based analysis. *Comput. Ind. Eng.* **2020**, *141*, 106282. [CrossRef]

31. Luo, D.; Guan, Z.; He, C.; Gong, Y.; Yue, L. Data-driven cloud simulation architecture for automated flexible production lines: Application in real smart factories. *Int. J. Prod. Res.* **2022**, *60*, 3751–3773. [CrossRef]
32. Ferreira, W.D.P.; Armellini, F.; Santa-Eulalia, L.A.D.; Thomasset-Laperrière, V. Extending the lean value stream mapping to the context of Industry 4.0: An agent-based technology approach. *J. Manuf. Syst.* **2022**, *63*, 1–14. [CrossRef]
33. Zhang, Y.; Wang, Y.; Wu, L. Research on Demand-driven Leagile Supply Chain Operation Model: A Simulation Based on AnyLogic in System Engineering. *Syst. Eng. Procedia* **2012**, *3*, 249–258. [CrossRef]
34. Ivanov, D. *Operations and Supply Chain Simulation with AnyLogic*; Berlin School of Economics and Law: Berlin, Germany, 2017.
35. Ahmed, A.; Page, J.; Olsen, J. Adopting Lean Six Sigma to AnyLogic Simulation in a Manufacturing Environment. In Proceedings of the 21st International Congress on Modelling and Simulation, Gold Coast, Australia, 29 November–4 December 2015.
36. Tiacci, L. Simultaneous balancing and buffer allocation decisions for the design of mixed-model assembly lines with parallel workstations and stochastic task times. *Int. J. Prod. Econ.* **2015**, *162*, 201–215. [CrossRef]
37. Scholl, A.; Boysen, N. Designing parallel assembly lines with split workplaces: Model and optimization procedure. *Int. J. Prod. Econ.* **2009**, *119*, 90–100. [CrossRef]
38. Akpınar, S.; Mirac Bayhan, G. A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. *Eng. Appl. Artif. Intell.* **2011**, *24*, 449–457. [CrossRef]
39. Roshani, A.; Fattahi, P.; Roshani, A.; Salehi, M.; Roshani, A. Cost-oriented two-sided assembly line balancing problem: A simulated annealing approach. *Int. J. Comput. Integr. Manuf.* **2012**, *25*, 689–715. [CrossRef]
40. Gulivindala, A.K.; Bahubalendruni, M.R.; Kumar Inkulu, A.; Varupala, S.V.P. A modified cut-set method for mechanical subassembly identification. *Assem. Autom.* **2021**, *41*, 659–680. [CrossRef]
41. Gulivindala, A.K.; Bahubalendruni, M.R.; Varupala, S.V.P. A heuristic method with a novel stability concept to perform parallel assembly sequence planning by subassembly detection. *Assem. Autom.* **2020**, *40*, 779–787. [CrossRef]
42. Lee, T.; Kim, Y.; Kim, Y. Two-sided assembly line balancing to maximize work relatedness and slackness. *Comput. Ind. Eng.* **2001**, *40*, 273–292. [CrossRef]
43. Li, Z.; Kucukkoc, I.; Janardhanan, M.N. Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem. *Comput. Oper. Res.* **2017**, *84*, 146–161. [CrossRef]
44. Becker, C.; Scholl, A. A survey on problems and methods in generalized assembly line balancing. *Eur. J. Oper. Res.* **2006**, *168*, 694–715. [CrossRef]
45. Aguilar, H.; García-Villoria, A.; Pastor, R. A survey of the parallel assembly lines balancing problem. *Comput. Oper. Res.* **2020**, *124*, 105061. [CrossRef]
46. Arora, S.; Barak, B. *Computational Complexity: A Modern Approach*; Cambridge University Press: Cambridge, UK, 2009; pp. 1–579. [CrossRef]
47. Katiraee, N.; Calzavara, M.; Finco, S.; Battaïa, O.; Battini, D. Assembly line balancing and worker assignment considering workers' expertise and perceived physical effort. *Int. J. Prod. Res.* **2022**. [CrossRef]
48. Bortolini, M.; Ferrari, E.; Gamberi, M.; Pilati, F.; Faccio, M. Assembly system design in the Industry 4.0 era: A general framework. *IFAC-Pap.* **2017**, *50*, 5700–5705. [CrossRef]
49. Cohen, Y.; Naseraldin, H.; Chaudhuri, A.; Pilati, F. Assembly systems in Industry 4.0 era: A road map to understand Assembly 4.0. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 4037–4054. [CrossRef]
50. Battini, D.; Finco, S.; Sgarbossa, F. *Human-Oriented Assembly Line Balancing and Sequencing Model in the Industry 4.0 Era*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 141–165. [CrossRef]
51. Fager, P.; Calzavara, M.; Sgarbossa, F. Modelling time efficiency of cobot-supported kit preparation. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 2227–2241. [CrossRef]
52. Peron, M.; Sgarbossa, F.; Strandhagen, J.O. Decision support model for implementing assistive technologies in assembly activities: A case study. *Int. J. Prod. Res.* **2022**, *60*, 1341–1367. [CrossRef]
53. Simonetto, M.; Peron, M.; Fragapane, G.; Sgarbossa, F. Digital Assembly Assistance System in Industry 4.0 Era: A Case Study with Projected Augmented Reality. In *Advanced Manufacturing and Automation X 10*; Springer: Singapore, 2021; pp. 644–651.
54. Dolgui, A.; Sgarbossa, F.; Simonetto, M. Design and management of assembly systems 4.0: Systematic literature review and research agenda. *Int. J. Prod. Res.* **2021**, *60*, 184–210. [CrossRef]
55. Stevanov, B.; Sremcev, N.; Lazarevic, M.; Anderla, A.; Sladojevic, S.; Vidicki, P. Optimization of the subassembly production process using simulation. *Int. J. Simul. Model.* **2022**, *21*, 633–674. [CrossRef]
56. Battini, D.; Delorme, X.; Dolgui, A.; Persona, A.; Sgarbossa, F. Ergonomics in assembly line balancing based on energy expenditure: A multi-objective model. *Int. J. Prod. Res.* **2016**, *54*, 824–845. [CrossRef]
57. Battaïa, O.; Otto, A.; Sgarbossa, F.; Pesch, E. Future trends in management and operation of assembly systems: From customized assembly systems to cyber-physical systems. *Omega* **2018**, *78*, 1–4. [CrossRef]
58. Vujica-Herzog, N.; Buchmeister, B.; Breznik, M. Ergonomics, digital twins and time measurements for optimal workplace design. In Proceedings of the AHFE International Conference, New York, NY, USA, 24–28 July 2022.
59. Alghazi, A.A. Balancing and Sequencing of Mixed Model Assembly Lines. Ph.D. Thesis, Clemson University, Clemson, SC, USA, 2017.
60. Scholl, A.; Becker, C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur. J. Oper. Res.* **2006**, *168*, 666–693. [CrossRef]

61. Schwarz, T.E.; Rubenbauer, W.; Rutrecht, B.; Pomberger, R. Forecasting Real Disassembly Time of Industrial Batteries based on Virtual MTM-UAS Data. In Proceedings of the 25th CIRP Life Cycle Engineering (LCE) Conference, Copenhagen, Denmark, 30 April–2 May 2018; pp. 927–931.
62. Schaub, K.; Caragnano, G.; Britzke, B.; Bruder, R. The European Assembly Worksheet. *Theor. Issues Ergon. Sci.* **2012**, *14*, 616–639. [CrossRef]
63. Caputo, F.; Greco, A.; Fera, M.; Macchiaroli, R. Digital twins to enhance the integration of ergonomics in the workplace design. *Int. J. Ind. Ergon.* **2019**, *71*, 20–31. [CrossRef]
64. Boysen, N.; Schulze, P.; Scholl, A. Assembly line balancing: What happened in the last fifteen years? *Eur. J. Oper. Res.* **2021**, *301*, 797–814. [CrossRef]
65. Maynard, H.B.; Stegemerten, G.J.; Schwab, J.L. *Methods-Time Measurement*; McGraw-Hill: New York, NY, USA, 1948; 292p.
66. Di Giornimo, G.; Di Martino, C.; Lanzotti, A.; Marzano, A.; Russo, G. Improving MTM-UAS to predetermine automotive maintenance times. *Int. J. Interact. Des. Manuf.* **2012**, *6*, 265–273. [CrossRef]
67. Nadworny, M.J. Frederick Taylor and Frank Gilbreth: Competition in Scientific Management. *Bus. Hist. Rev.* **1957**, *31*, 23–34. [CrossRef]
68. Vincent, S.; Law, A.M. ExpertFit: Total support for simulation input modeling. In Proceedings of the 1995 Winter Simulation Conference Proceedings, Arlington, VA, USA, 28 February 1995; pp. 395–400.
69. Buchmeister, B. *Modelling and Simulation of Production Systems*; DAAAM International Publishing: Vienna, Austria, 2020; 140p.
70. Sabuncuoglu, I.; Erel, E.; Tanyer, M. Assembly line balancing using genetic algorithms. *J. Intell. Manuf.* **2000**, *11*, 295–310. [CrossRef]

applied sciences

MDPI

Communication

Density-Based Prioritization Algorithm for Minimizing Surplus Parts in Selective Assembly

Kanghyeon Shin [1] and Kyohong Jin [2,*]

[1] ICT Convergence Center, Changwon National University, Changwon 51140, Republic of Korea; sheen98@changwon.ac.kr
[2] Department of Electronic Engineering, Changwon National University, Changwon 51140, Republic of Korea
* Correspondence: khjin@changwon.ac.kr; Tel.: +82-55-213-3656

Abstract: Selective assembly is a manufacturing method that matches and assembles pairs of parts in a manner that offsets the machining errors of these parts. In the production of products requiring high precision and efficient mass production, flow production and search-based selective assembly must be combined for market competitiveness; however, this method increases computational costs and generates many surplus parts. Therefore, research should aim to minimize surplus parts in search-based selective assembly at a low computational cost to suit flow production systems. In this paper, we propose the density-based prioritization (DBP) algorithm, which minimizes surplus parts in the search-based selective assembly of flow production systems. In addition, a method of varying the assembly tolerance is developed and incorporated into DBP to increase its process capability. The proposed algorithm requires an assembly facility to prepare parts with as many different sizes as possible. This paper confirms that DBP reduces computational costs and surplus parts while enhancing process capability.

Keywords: prioritization; selective assembly; surplus part; flow production; ball bearing

1. Introduction

Selective assembly is a manufacturing method that measures the machining errors of processed parts and then matches and assembles specific pairs such that these errors are offset. Consequently, high-precision assemblies can be obtained even with low-precision parts, enabling the mass production of precision products. However, due to the difficulty of achieving an ideal combination of all parts according to dimensional distribution of parts or the selective assembly method, parts without mates (surplus parts) remain. The occurrence of surplus parts wastes manufacturing resources and increases manufacturing costs. Therefore, minimizing surplus parts is important for selective assembly [1].

Various methods of selective assembly have been studied according to the characteristics of fabricated products and assembly facilities. Raj et al. [2] developed an algorithm based on particle swarm optimization to minimize surplus parts in selective assembly that must satisfy multiple assembly tolerances. In addition, Raj et al. [3] proposed a method that used the non-dominated sorting genetic algorithm II to optimize assembly precision and eliminate surplus parts. Asha and Babu [4] applied genetic, simulated annealing, and memetic algorithms to selective assembly to compare assembly precision and the number of surplus parts. Filipovich and Kopp [5] modified a selective assembly model based on a parameter estimation algorithm to reduce sorting errors due to measurement errors. Aderiani et al. [6] proposed the use of a genetic algorithm to improve assembly precision under any distribution of parts without producing surplus parts. Furthermore, Aderiani et al. [7] improved the phenotype–genotype mapping method used with evolutionary optimization algorithms for selective assembly to accelerate optimization. Liu et al. [8] proposed a method that used a fireworks algorithm to optimize assembly precision and the number of surplus parts in multi-matching selective assembly under a non-normal dimensional distribution of parts. Kannan and Pandian [9,10] proposed a selective assembly model that used a genetic algorithm to minimize surplus parts within a strict assembly tolerance.

These selective assembly approaches can be divided into group- and search-based methods. Group-based methods classify parts to be assembled with each other into groups according to their dimensions and organize groups so that machining errors are offset, whereas search-based methods identify and assemble the best combinations of parts through calculation. Group-based methods have simple structures and fast processing speeds, enabling efficient mass production, but when the required precision is high, the facility should be larger so that groups can be further subdivided. Search-based methods enable more precise assembly than group-based methods. Search-based methods require calculation every time a new component is introduced, so studies on these approaches focus on batch production systems rather than flow production systems. However, as the precision parts in automobiles and industrial machineries require very high precision and efficient mass production, flow production systems and search-based selective assembly must be combined for market competitiveness. Therefore, research should focus on minimizing surplus parts in search-based selective assembly at a low computational cost to suit flow production systems.

In search-based selective assembly, a dimensional concentration phenomenon occurs in which the supply and assembly frequencies of parts according to size become unbalanced. As this phenomenon intensifies, the diversity of the parts decreases, thus reducing the probability that the combinations satisfy the assembly tolerance. In this paper, we propose the density-based prioritization (DBP) algorithm to minimize surplus parts in the search-based selective assembly of flow production systems. DBP regards the similarity of a part with the other parts as density on the dimensional coordinate and gives high selection priority to parts with high density to balance the supply and assembly frequencies. We examine the selective assembly procedure for producing precision ball bearings and analyze the factors that cause surplus parts to occur. We then evaluate selective assembly with DBP and compare it with traditional selective assembly.

2. Search-Based Selective Assembly of Flow Production Systems and Surplus Parts

2.1. Selective Assembly Procedure

In this study, an actual precision single-row deep-groove ball bearing assembly process is analyzed as an example of selective assembly. The ball bearing consists of an outer ring, an inner ring, and balls, as shown in Figure 1. In this example, the dimensions of each part and the specifications of the assembly clearance are as follows:

- Outer ring raceway diameter: $A = 40^{+0.024}_{-0.005}$ mm;
- Inner ring raceway diameter: $B = 24 \pm 0.025$ mm;
- Ball diameter: $C = 8 \pm 0.0005$ mm;
- Assembly clearance: $Y = 0.009 \pm 0.0025$ mm.

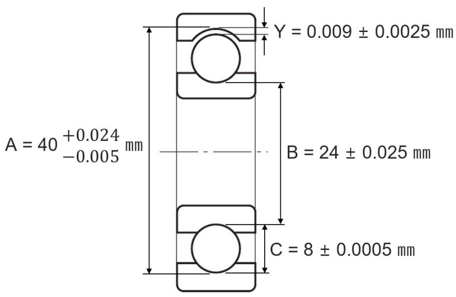


Figure 1. Precision ball bearing assembly.

The assembly clearance is formed according to the dimensions of each part as follows:

$$Y = A - B - 2C. \tag{1}$$

The given ball bearing assembly process consists of the structure shown in Figure 2. The outer and inner rings are supplied one by one in every process cycle by the flow production system and the balls are preproduced and placed in seven tanks. The assembly facility has thirty slots for storing the outer rings one by one. Additional types of balls are prepared to increase the possibility of assembly between the outer and inner rings. Seven types of balls have different biases, ranging from −6 to +6 μm in increments of 2 μm, with respect to the ball diameter of 8 mm, and the tolerance is ±0.5 μm, as ever. When an inner ring is supplied, the assembly facility checks the stored outer rings and balls to find a combination that can satisfy the assembly clearance tolerance. According to Equation (1), in a total of 210 cases, assembly is performed by selecting the combination whose assembly clearance is the most approximate to 9 μm and the slot vacated by the selected outer ring is refilled with a newly supplied outer ring.

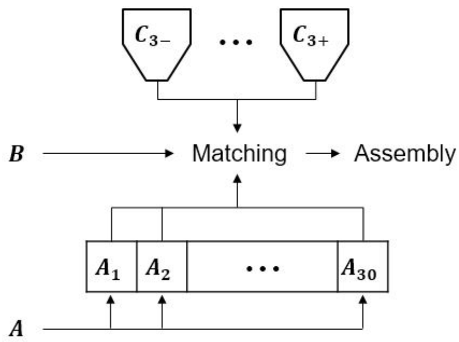


Figure 2. Selective assembly of precision ball bearing.

The diameters of each ball in the tanks cannot be identified and recorded, thus all balls in the same tank are deemed to have the same diameter. Therefore, the allowable range of Y should be set to ±1.2 μm in consideration of the tolerance (±1 μm) and other measurement errors. In addition, if the center value of each dimension in Equation (1) were removed, the calculation would only be possible with the error values. Based on this, the dimensional values to be used for selective assembly are redefined as follows:

- Outer ring raceway diameter error: $A = \pm 15$ μm;
- Inner ring raceway diameter error: $B = \pm 25$ μm;
- Ball diameter bias:

$$C_{3-} = -6 \ \mu\text{m}$$
$$C_{2-} = -4 \ \mu\text{m}$$
$$C_{-} = -2 \ \mu\text{m}$$
$$C_{0} = 0 \ \mu\text{m}$$
$$C_{+} = 2 \ \mu\text{m}$$
$$C_{2+} = 4 \ \mu\text{m}$$
$$C_{3+} = 6 \ \mu\text{m};$$

- Assembly clearance tolerance: $Y = \pm 1.2$ μm.

Then, the combination whose assembly clearance is the closest to the target value is identified by finding the combination where Y is the most approximate to 0 μm.

If no combination satisfies the assembly tolerance, the outer rings in all slots will be removed and all slots will be refilled with newly supplied outer rings. The inner ring remains in the assembly facility and will be used again when assembly resumes. Therefore, the inner rings supplied to the assembly facility as good products must be assembled. The extracted outer rings are reprocessed or discarded as surplus parts. In addition, assembly is paused until all slots are refilled, so the generation of surplus parts reduces process productivity and increases production costs.

2.2. Cause of Surplus Parts

If even one slot has an outer ring that satisfies the assembly tolerance, no surplus part will remain. Since the assembly facility does not know which diameter inner rings will be supplied next, the diameters of the outer rings in the slots should be as varied as possible so that at least one outer ring can be assembled regardless of which diameter inner ring is supplied. Figure 3 shows the dimensional distribution of outer ring raceway diameter errors in the slots over time from some of the data obtained from the assembly facility to illustrate how the distribution changes as the assembly process proceeds. Over time, the diameter errors gradually converge to similar values. As this dimensional concentration phenomenon intensifies, the diversity of the outer rings decreases, thus reducing the probability that the combinations satisfy the assembly tolerance. This phenomenon occurs because the dimensional distribution of the outer and inner ring raceway diameter errors becomes unbalanced as the machine tools undergo constant wear and adjustment. However, changes in this distribution are difficult to control precisely in the machine tools. Therefore, a selection strategy should be developed to balance the supply and assembly frequencies throughout the range of the outer ring raceway diameter.

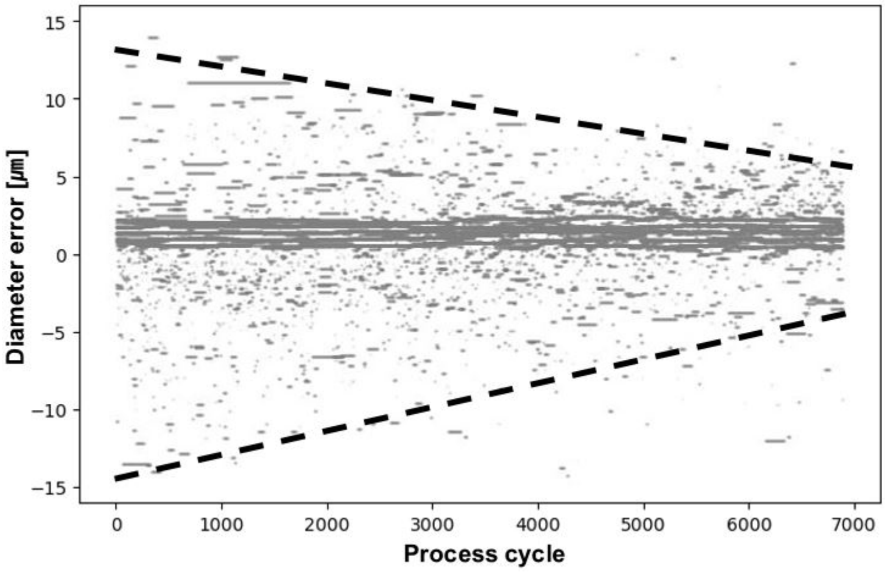


Figure 3. Dimensional distribution of outer ring raceway diameter errors in slots over time.

3. DBP Algorithm

As confirmed in Section 2.2, the frequencies of supply and assembly must be balanced throughout the range of measurements to minimize surplus parts in the search-based selective assembly of flow production systems. Accordingly, we propose the DBP algorithm, which prioritizes the selection of parts with many other similar sized parts in the slots.

DBP regards the similarity of a part with the other parts for each slot as linear density on the dimensional coordinate; these slots are prioritized in order of density. This linear density λ can be understood as a quantity Q of parts per unit range L of measurement; this is equal to the inverse of the average distance *E(D)* between neighboring parts on the dimensional coordinate:

$$\lambda = \frac{Q}{L} = \frac{1}{E(D)}. \quad (2)$$

Thus, DBP obtains the density for one part λ_i as the inverse of the average distance from one part x_i to its two nearest parts in each of the smaller and larger sides on the dimensional coordinate, x_{i-1} and x_{i+1}:

$$\lambda_i = \frac{1}{\frac{1}{2}\{(x_i - x_{i-1}) + (x_{i+1} - x_i)\}} = \frac{2}{x_{i+1} - x_{i-1}}. \quad (3)$$

In addition, the order of density is the reverse of the order of average distance and the order of average distance is the order of sum of the two distances:

$$\lambda_i > \lambda_j \Leftrightarrow x_{i+1} - x_{i-1} < x_{j+1} - x_{j-1}. \quad (4)$$

Therefore, unnecessary calculations for prioritization are omitted. Here, since the smallest and largest parts have only one side which is smaller or larger, the distances of these parts from their nearest parts are multiplied by two. The dimensional concentration phenomenon is alleviated by determining whether the outer rings can be assembled in order of priority so that they can be selected first. The implementation of the DBP algorithm is as follows:

Algorithm: Density-Based Prioritization

Input: Array of measurements of parts X
Output: Array of sorted indexes by priority P
1. n ← length (X)
2. initialize D[1 . . . n]
3. P ← [1 . . . n]
4. sort X and P by X in ascending
5. **for** i ← 2 . . . n−1 **do**
6. D[i] ← X[i+1]−X[i−1]
7. **end**
8. D[1] ← 2 × (X[2]−X[1])
9. D[n] ← 2 × (X[n]−X[n−1])
10. sort P by D in ascending
11. return P

The selective assembly process selects the outer ring with the highest priority set by DBP among the outer rings that can be assembled. However, this is a poor strategy in terms of process capability, which is an indicator of how well the precision of the process results meets the tolerances required by the process. Process capability should be controlled in the process of manufacturing precision products. The process capability C_{pk} is calculated as follows:

$$C_{pk} = \min\left[\frac{USL - \hat{\mu}}{3\hat{\sigma}}, \frac{\hat{\mu} - LSL}{3\hat{\sigma}}\right]. \quad (5)$$

where USL is the upper specification limit, LSL is the lower specification limit, $\hat{\mu}$ is the mean of the process, and $\hat{\sigma}$ is the variability of the process; the sigma levels corresponding to different C_{pk} values are shown in Table 1 [11].

Table 1. Process capabilities and sigma levels.

C_{pk}	Sigma Level
2	6σ
1.67	5σ
1.33	4σ
1	3σ
0.67	2σ

Even if all parts meet the specifications, a low process capability indicates a high probability of producing defective products due to future fluctuations. Therefore, a method of reducing the process deviation is needed. A phasing method can be implemented to determine whether or not products can be assembled in a narrower tolerance and again in the original tolerance if there are no parts that can be assembled. Since the DBP-applied selective assembly system does not consider the assembly precision, the distribution of the assembly results will approximate a uniform distribution within the tolerance. Therefore, the process capability is estimated as follows:

$$\hat{\sigma} \approx \sigma_{uniform} = \frac{UTL - LTL}{\sqrt{12}}, \tag{6}$$

$$\hat{\mu} \approx \mu_{uniform} = \frac{LTL + UTL}{2}, \tag{7}$$

$$\frac{LTL + UTL}{2} = \frac{LSL + USL}{2} \Rightarrow C_{pk} = \frac{USL - LSL}{6\hat{\sigma}}, \tag{8}$$

$$\therefore C_{pk} \approx \frac{(USL - LSL)\sqrt{12}}{6(UTL - LTL)}. \tag{9}$$

where *UTL* is the upper tolerance limit and *LTL* is the lower tolerance limit. The assembly tolerance satisfying a specific process capability is estimated as follows:

$$UTL \approx \frac{LSL + USL}{2} + \frac{(USL - LSL)}{C_{pk}\sqrt{12}},\ LTL \approx \frac{LSL + USL}{2} - \frac{(USL - LSL)}{C_{pk}\sqrt{12}}. \tag{10}$$

4. Performance Evaluation and Results

For evaluating the effectiveness of DBP, an assembly scenario was reproduced using data collected from the actual precision ball bearing assembly facility described in Section 2.1 and the DBP algorithm was simulated. The simulation had a total of 125,447 cycles and one outer ring and one inner ring were supplied for each cycle. Thirty outer ring slots and seven ball tanks were used. The effectiveness of DBP was assessed by comparing it with the traditional algorithm. Three versions of DBP with different levels of tolerance phasing were created. The compared algorithms were as follows:

1. Traditional: select the combination where Y is the most approximate to 0 μm among combinations where Y is within ±1.2 μm.
2. DBP-I: select the combination with the highest-priority outer ring (set by DBP) among combinations where Y is within ±1.2 μm.
3. DBP-II: Select the combination with the highest-priority outer ring (set by DBP) among combinations where Y is within ±0.6 μm. If no satisfactory combination is identified, explore using ±1.2 μm.
4. DBP-III: Select the combination with the highest-priority outer ring (set by DBP) among combinations where Y is within ±0.4 μm. If no satisfactory combination is identified, explore using ±0.8 and ±1.2 μm in sequence.

Table 2 shows the surplus part ratio and process capability for each algorithm in the simulation. The surplus part ratio is the ratio of the surplus outer rings to the total outer ring supply; the reduction rate compared with the traditional algorithm is the difference in the number of the surplus outer rings between the traditional and applied algorithms divided by the number of the surplus outer rings in the traditional algorithms. Findings confirm that surplus parts can be reduced using the DBP algorithm compared with the traditional algorithm. DBP-I does not generate any surplus parts but reduces C_{pk} to 1.109. As the assembly tolerance is phased under DBP-II and DBP-III, the surplus parts gradually increase, but they are better than the results of the traditional algorithm; C_{pk} is also improved. Accordingly, an appropriate assembly clearance tolerance can be set by balancing the surplus part ratio and process capability. Figure 4 shows the dimensional distributions of the outer ring diameter errors in the slots for each DBP version. The dimensional concentration phenomenon is less than that in Figure 3. Compared with DBP-I, DBP-II and DBP-III show a weaker alleviation of the dimensional concentration phenomenon as the tolerance is phased. This is because DBP operates more similarly to the traditional algorithm as the tolerance is phased. If the tolerance were phased in units of 0.1 μm, which is the minimum unit, DBP would produce exactly the same results as the traditional algorithm. Therefore, the surplus part ratio and process capability are adjusted according to the level of tolerance phasing.

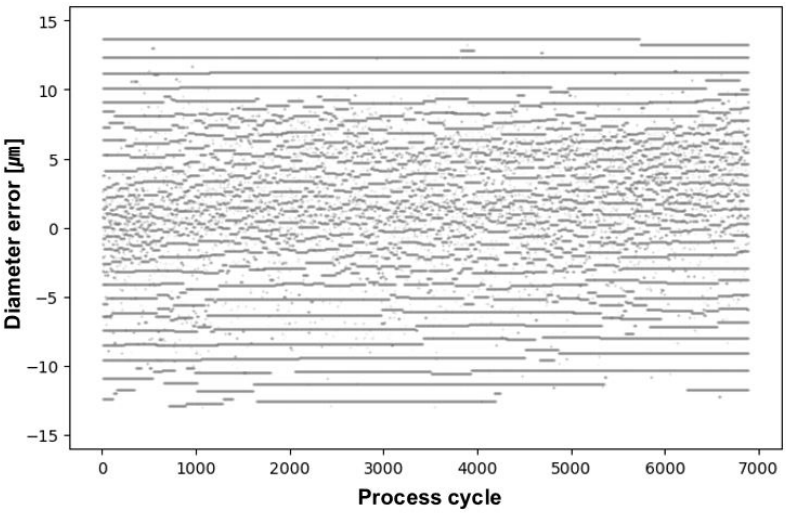
Table 2. Surplus part ratios and process capabilities of compared algorithms.

	Surplus Part Ratio (%)	Reduction Rate Compared with Traditional (%)	C_{pk}
Traditional	0.806	-	1.877
DBP-I	0.000	100.000	1.109
DBP-II	0.033	95.906	2.106
DBP-III	0.132	83.623	2.933

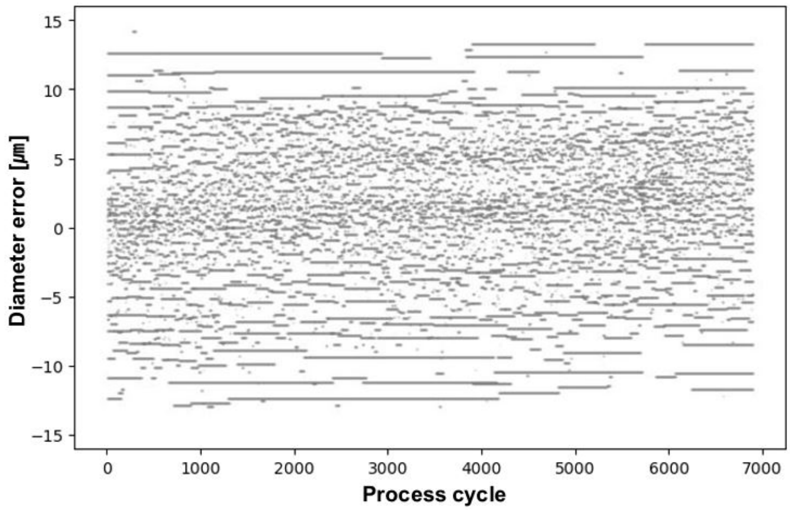
Table 3 shows the operation times of the algorithms to evaluate their computational costs. The operation time is measured starting from the completion time of the inner ring raceway diameter measurement and ending at the decision of one combination of the parts to be assembled. The traditional algorithm always examines all combinations, so it shows a constant operation time of approximately 210 μs/cycle. DBP does not have to examine the subordinated combinations if products can be assembled into high-priority combinations, so its operation time varies in some cases. The minimum operation time is approximately 50 μs/cycle, regardless of the phase of the assembly tolerance, and the maximum operation time increases as the assembly tolerance is phased in more detail. The average operation time of DBP over the entire period is shorter than that of the traditional algorithm. As a result, DBP did not delay the assembly process. However, the dynamic operation time can destabilize the process cycle time and pose a potential risk factor for the entire production system. Therefore, a proper buffer should be placed immediately after the assembly process to introduce DBP.

Table 3. Operation times of compared algorithms.

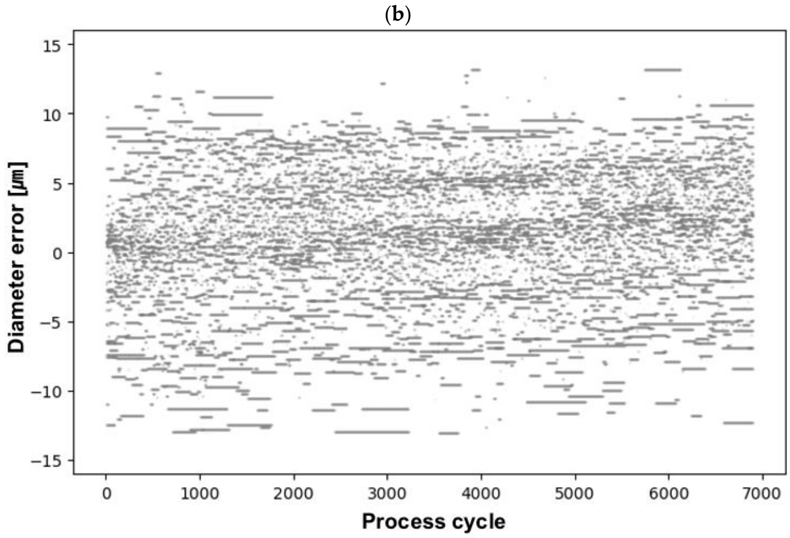
	Minimum Operation Time (μs/Cycle)	Maximum Operation Time (μs/Cycle)	Average Operation Time (μs/Cycle)
Traditional	210.2	210.6	210.4
DBP-I	50.94	439.3	72.56
DBP-II	50.79	674.8	97.96
DBP-III	50.33	885.2	128.4
CPU Intel Core i9-10940X 3.30 GHz		OS Windows 10 Pro	
RAM 4 × 32 GB DDR4 2666 MHz		Language Python 3.11.0	



(a)



(b)



(c)

Figure 4. Dimensional distribution of outer ring diameter errors in slots over time: (**a**) DBP-I; (**b**) DBP-II; (**c**) DBP-III.

5. Conclusions

In this paper, we propose the DBP algorithm, which minimizes the surplus parts in the search-based selective assembly of flow production systems. DBP evaluates the density of a part (similarity of a part with the other parts); then, for assembly, DBP prioritizes the parts in order of density. The proposed algorithm alleviates the dimensional concentration phenomenon by allowing high-priority parts to be selected preferentially. In addition, tolerance phasing is presented to solve the process capability degradation of DBP.

For assessing the effectiveness of DBP, an assembly scenario was reproduced using data collected from an actual precision ball bearing assembly facility. Three versions of DBP with different levels of tolerance phasing were created and were compared with the traditional algorithm through simulation. Their surplus part ratios, process capabilities, and operation times were analyzed and compared. Results confirmed that DBP could reduce surplus parts and improve process capability compared with the traditional algorithm by setting an appropriate phase of assembly tolerance. In addition, the average operation time of DBP over the entire period was shorter than that of the traditional algorithm. However, since the dynamic operating time can destabilize the process cycle time, a proper buffer must be placed immediately after the assembly process to introduce DBP. In this study, only one assembly scenario was used to assess the effectiveness of DBP and the influence of various factors that may occur in the actual factory were not considered. Therefore, further analysis of data for various scenarios and empirical works is needed to prove the practical applicability of the DBP.

Author Contributions: Conceptualization, K.S. and K.J.; methodology, K.S. and K.J.; software, K.S.; validation, K.S. and K.J.; formal analysis, K.S. and K.J.; investigation, K.S. and K.J.; resources, K.J.; data curation, K.S.; writing—original draft preparation, K.S. and K.J.; writing—review and editing, K.J.; visualization, K.S.; supervision, K.J.; project administration, K.J.; funding acquisition, K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2023-2016-0-00318) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data were obtained from NSK Korea Co., Ltd. (Changwon Plant) and are available from the authors with the permission of NSK Korea Co., Ltd. (Changwon Plant).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mansoor, E.M. Selective assembly—Its analysis and applications. *Int. J. Prod. Res.* **1961**, *1*, 13–24. [CrossRef]
2. Raj, M.V.; Sankar, S.S.; Ponnambalam, S.G. Maximising Manufacturing System Efficiency for Multi-Characteristic Linear Assembly by Using Particle Swarm Optimisation in Batch Selective Assembly. *Int. J. Prod. Res.* **2011**, *49*, 6491–6516. [CrossRef]
3. Raj, M.V.; Sankar, S.S.; Ponnambalam, S.G. Minimizing Clearance Variations and Surplus Parts in Multiple Characteristic Radial Assembly through Batch Selective Assembly. *Int. J. Adv. Manuf. Technol.* **2011**, *57*, 1199–1222. [CrossRef]
4. Asha, A.; Babu, J.R. Comparison of Clearance Variation Using Selective Assembly and Metaheuristic Approach. *Int. J. Latest Trends Eng. Technol.* **2017**, *8*, 148–155. [CrossRef]
5. Filipovich, O.; Kopp, V. Simulation Model of Selective Assembly of Two Parts with Sorting by the Estimated Values. *MATEC Web Conf.* **2018**, *224*, 01114. [CrossRef]
6. Aderiani, A.R.; Wärmefjord, K.; Söderberg, R. A Multistage Approach to the Selective Assembly of Components without Dimensional Distribution Assumptions. *J. Manuf. Sci. Eng.* **2018**, *140*, 071015. [CrossRef]
7. Aderiani, A.R.; Wärmefjord, K.; Söderberg, R. An Improved Phenotype-Genotype Mapping for Solving Selective Assembly Problem Using Evolutionary Optimization Algorithms. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 061010. [CrossRef]
8. Liu, Z.; Nan, Z.; Qiu, C.; Tan, J.; Zhou, J.; Yao, Y. A Discrete Fireworks Optimization Algorithm to Optimize Multi-Matching Selective Assembly Problem with Non-Normal Dimensional Distribution. *Assem. Autom.* **2019**, *39*, 323–344. [CrossRef]
9. Kannan, S.M.; Pandian, G.R. A New Selective Assembly Model for Achieving Specified Tolerance in High Precision Assemblies. *Int. J. Precis. Eng. Manuf.* **2020**, *21*, 1217–1230. [CrossRef]

10. Kannan, S.M.; Pandian, G.R. A New Selective Assembly Model for Achieving Specified Clearance in Radial Assembly. *Mater. Today Proc.* **2021**, *46*, 7411–7417. [CrossRef]
11. Kane, V.E. Process Capability Indices. *J. Qual. Technol.* **1986**, *18*, 41–52. [CrossRef]

applied sciences

MDPI

Article

Robustness Optimization of Cloud Manufacturing Process under Various Resource Substitution Strategies

Xiaodong Zhang [1], Xin Zheng [1,*] and Yiqi Wang [2]

1 School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China; xdzhang@manage.ustb.edu.cn

2 China Academy of Launch Vehicle Technology, Beijing 100076, China; yqwang_17@163.com

* Correspondence: b20200433@xs.ustb.edu.cn

Abstract: Cloud manufacturing is characterized by large uncertainties and disturbances due to its networked, distributed, and loosely coupled features. To target the problem of frequent cloud resource node failure, this paper proposes (1) three resource substitution strategies based on node redundancy and (2) a new robustness analysis method for cloud manufacturing systems based on a combination of the complex network and multi-agent simulation. First, a multi-agent simulation model is constructed, and simulation evaluation indexes are designed to study the robustness of the dynamic cloud manufacturing process (CMP). Second, a complex network model of cloud manufacturing resources is established to analyze the static topological robustness of the cloud manufacturing network. Four types of node failure modes are defined, based on the initial and recomputed topologies. Further, three resource substitution strategies are proposed (i.e., internal replacement, external replacement, and internal–external integration replacement) to enable the normal operation of the system after resource node failure. Third, a case study is conducted for a cloud manufacturing project of a new energy vehicle. The results show that (1) the proposed robustness of service index is effective at describing the variations in CMP robustness, (2) the two node failure modes based on the recalculated topology are more destructive to the robustness of the CMP than the two based on the initial topology, and (3) under all four failure modes, all three resource substitution strategies can improve the robustness of the dynamic CMP to some extent, with the internal–external integration replacement strategy being most effective, followed by the external replacement strategy, and then the internal replacement strategy.

Keywords: cloud manufacturing; robust optimization; resource substitution; complex network; multi-agent simulation

1. Introduction

In the era of Industry 4.0, with the rapid development of Internet technology, information technology, and manufacturing technology, the traditional large-scale manufacturing mode is gradually being replaced by customized service modes. A series of advanced networked manufacturing modes, such as application service providers, manufacturing grids, agile manufacturing, and global manufacturing have been proposed successively [1]. In this context, Li et al. [2] introduced the concept of "cloud manufacturing"—a new service-oriented networked manufacturing mode that gathers manufacturing resources and capabilities together on the cloud platform, escaping the limitations of space and distance; through service integration, the sharing of manufacturing resources and capabilities is fully realized [3]. Since its inception, cloud manufacturing has attracted widespread attention because of its advanced ideas and technical concepts [4].

The cloud manufacturing mode extends the manufacturing environment to multiple user subjects, service subjects, and geographical spaces. As such, it faces a high level of uncertainty and disturbance [5]. The cloud manufacturing system (CMS) can reduce the impact of some disturbances (e.g., demand change, order fluctuation, and emergency order insertion) through its own flexible configuration and self-adjustment strategies, but relatively serious disturbances can cause a variety of damage, such as insufficient supervision of the cloud platform, random the withdrawal of cloud service providers from the platform, the failure of cloud resource nodes, and the interruption of cloud paths, among others [6]. Therefore, it is of great practical significance for the implementation and deployment of cloud manufacturing projects to accurately identify the impact of uncertain environments on cloud manufacturing [7–9], explore the robustness level of the cloud manufacturing process (CMP) under different interference modes, and formulate corresponding robustness recovery strategies to improve the stability and anti-interference of the system [10–12].

2. Related Works

2.1. Robustness Analysis Study of Manufacturing Systems

Many scholars have conducted research on the robustness of advanced manufacturing systems and networks. Gao et al. [13] applied complex network theory to the manufacturing industry, proposing the construction method of the complex network failure model. Yana et al. [14] proposed that the vulnerability of the manufacturing network may lead to new risks; they analyzed the topology structure and vulnerability of the cloud manufacturing network (CMN) and put forward management suggestions. Cauvin et al. [15] analyzed the impacts of interruption events on the entire industrial system and proposed a cooperative repair approach based on a distributed industrial system to limit these impacts. Li et al. [16] established a collaborative manufacturing services network model using the complex network method, defined six fault types, and proposed corresponding fault detection methods. Further, the cascaded propagation characteristics of different faults were revealed, and corresponding control strategies were proposed. Lachenmaier et al. [17] analyzed the changing individualized requirements, risks, and possible solutions in cyber-physical systems. Galaske and Anderl [18] proposed a decision support method for the terminal management process of a cyber-physical production system, modeling and simulating respective scenarios of interruption events and response strategies. Hou [19] established the process relationship network of the flexible job shop by analyzing the relationships among basic production factors, such as machines and workpieces in the shop. They proposed adaptive preventive maintenance and buffer time insertion strategies. Zhang [20] proposed the definition of the manufacturing product assurance network; they established an evolutionary model of the manufacturing product assurance weighted network on this basis, and then analyzed the robustness characteristics of the network.

Such research demonstrates that cloud manufacturing and other advanced networked manufacturing modes have large uncertainties and risks due to their networked, distributed, and loosely coupled characteristics. Robustness analysis of these modes has become a popular issue in academic circles, with many scholars exploring this from a variety of perspectives.

2.2. Robustness Enhancement Strategies and Recovery Measures

The specific robustness enhancement and defense strategies vary from network to network, but from the perspective of a network structure, establishing redundant nodes or links is one of the most common methods to improve network robustness. As early as 2005, Beygelzimer et al. [21] proposed that people are usually faced with the problem of improving the robustness of an existing network that cannot be substantially modified or redesigned: only minor modifications are allowed, such as adding new nodes, reconnecting partial edges, or adding new edges. Ma [22] stated that the robustness of the whole network can be improved by increasing the number of reliable nodes in the network or improving the reliability of a small number of nodes in the network. Ji et al. [23] argued that it is impossible to enhance the reliability of every node in the network, so it is instead necessary to specify the priority of nodes and links and focus on their protection.

Regarding the study of reliable nodes, Li [24] proposed a heuristic scheduling rule based on controlling critical nodes for the scheduling optimization of the job shop in a disturbed environment, and Wang [25] used both the improved node shrinkage method and the triangular fuzzy number method to comprehensively evaluate the importance of nodes in the network, proposing corresponding improvement measures in terms of critical node protection. Regarding the study of adding links (i.e., connected edges), Ji et al. [23] proposed two new strategies: low inter degree–degree difference addition and random inter degree–degree difference addition. They verified the effectiveness of these proposed strategies through a comparison with four existing link-addition strategies (i.e., random addition, low degree addition, low betweenness addition, and algebraic connectivity-based addition). Wang et al. [26] proposed a preferred connectivity strategy based on the structure of interdependent networks. They applied this strategy to three existing link-addition strategies (i.e., random addition, low degree addition, and low inter degree–degree difference addition), finding that each improved strategy was significantly better than the previous one in terms of robustness improvement. The effects of link-addition strategies on the improvement of the robustness of different networks have been further investigated [27–30].

Measures such as adding redundant nodes and connected edges can effectively improve network robustness. As such, these measures are receiving increasing attention from scholars, and they also provided ideas for the design of the cloud manufacturing-based robustness improvement strategy in this paper. Targeting the problem of frequent cloud resource node failure, this paper attempts to establish redundant nodes so that appropriate alternative resources [31] can be used to avoid the complete breakdown of the manufacturing process or system when the original manufacturing resource nodes fail. Based on this and the background characteristics of cloud manufacturing, this paper presents three kinds of robustness improvement strategies: internal resource replacement, external resource replacement, and internal–external integration replacement.

2.3. Multi-Agent Simulation Study of Cloud Manufacturing System

The existing research on the robustness of advanced manufacturing systems mostly uses the complex network analysis method, which is well suited to reflect the structural characteristics of the system. As a type of networked manufacturing mode, the structural characteristics of cloud manufacturing is a key topic in robustness research; however, the dynamic operation process, logical judgment, and dynamic temporal relationship among entities are also aspects of cloud manufacturing that should be focused on.

The CMS contains a variety of entities, and various forms of behavior interaction and information transfer exist between the same entities and different entities, showing the characteristics of the complex system. Analyzing the complexity through simple, intelligent, and autonomous entities, such as agents in multi-agent systems, is considered an appropriate approach to address this challenge in industrial scenarios [32]. Further, the multi-agent simulation modeling of the CMS is presently a popular research topic. For example, Zhao et al. designed and implemented an agent-based cloud manufacturing simulation platform, where the simple reflective agent was used to encapsulate the resources and the complex agent was used to encapsulate the services. This gave the cloud platform a five-layer architecture (i.e., the data layer, low tool layer, management layer, upper tool layer, and application layer). Dobrescu et al. [33] proposed a cloud simulation platform to provide computing resources and services for the hybrid simulation of virtual manufacturing systems, and Chen and Chiu [34] developed a cloud-based factory simulation experiment system. Zhang et al. [35] analyzed typical smart manufacturing simulation techniques from three aspects: manufacturing unit simulation, manufacturing integration simulation, and manufacturing intelligence simulation. In addition, some scholars have conducted multi-agent simulation modeling research on cloud manufacturing from multiple perspectives, such as cloud service entity encapsulation [36,37], selection and scheduling [38–40], and trust and security issues [5,8], among others. Multi-agent simulation has become an important tool for cloud manufacturing research, yet current multi-agent simulation research on the robustness of cloud manufacturing appears relatively rare.

This paper proposes a robustness analysis method that combines the complex network and multi-agent simulation to investigate the optimization and enhancement of CMS robustness under multiple resource substitution strategies. The complex network perspective can reflect the structural robustness of CMSs, while multi-agent simulation can consider the process robustness of CMSs from multiple dimensions, such as time, cost, quality, and reliability. The combination of these two perspectives extends the robustness analysis object of the CMS from the CMN to the CMP, thereby realizing the dual-dimensional analysis of the static structure and dynamic process of CMS robustness.

The rest of this paper is arranged as follows. Section 3 constructs a multi-agent simulation model of the CMP and proposes process robustness indexes from the simulation perspective. Section 4 establishes a complex network model of cloud manufacturing resources and selects structural robustness metrics from the network perspective. Section 5 defines four types of resource node failure modes and three types of resource replacement strategies to deal with resource failure. Section 6 conducts a case study, combining multi-agent simulation software Anylogic and Python 3.0 tools to study the changes in the robustness of cloud manufacturing under different failure modes. Section 7 provides the research conclusions and future prospects.

3. Model of the Cloud Manufacturing Process and Robustness Evaluation Indicator Based on Multi-Agent Simulation

3.1. Construction of Multi-Agent Simulation Model

The cloud platform, cloud task, cloud resource, cloud message, cloud order, and other types of subjects are all contained in the CMS, along with two different types of user roles: cloud service providers and cloud demanders [41]. The CMP [2] basically entails the following, as indicated in Figure 1:

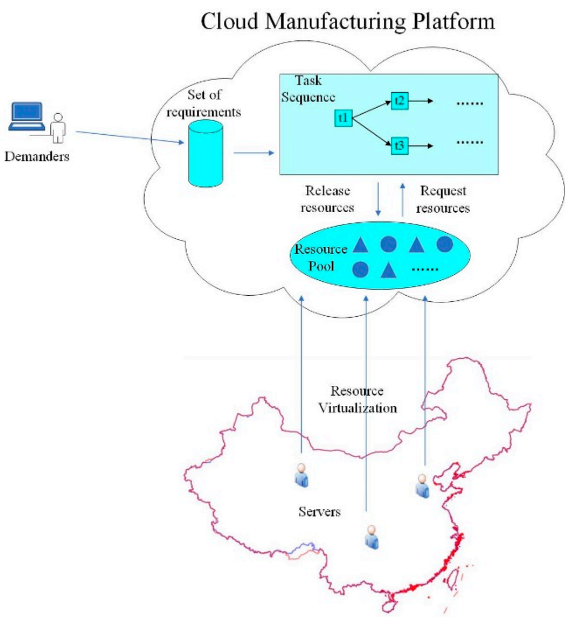


Figure 1. Schematic diagram of CMP.

(1) Through information transformation, resource sensing, resource access, unified modeling of cloud services, and other technologies, cloud service providers integrate different kinds of manufacturing equipment and manufacturing capability resources into the cloud platform and deposit them into the cloud resource pool. This allows globally distributed resources to be managed and shared centrally, thereby circumventing the spatial and geographical limitations.

(2) Utilizing terminal devices, cloud demanders submit their requests for services (i.e., orders) to the cloud platform. The cloud demand set uniformly stores orders awaiting processing from various cloud demanders.
(3) In accordance with the service route of the to-be-processed order, the cloud platform integrates and adapts various cloud tasks to create structured and reliable cloud task sequences.
(4) In order to perform cloud manufacturing services, the platform imports each order into the appropriate cloud task sequence when the cloud demand set is not vacant. Based on the task type, the appropriate resources are requested from the resource pool while processing cloud tasks. After being requested, resources in an inactive state transition to a busy state. The resource is released and returned to an idle state once the assignment has been finished.

Multiple entity types are present in the CMS, and numerous forms of information transmission and behavior interactions occur among entities of the same and different types. Consequently, the CMS model can be stated as follows:

$$\boldsymbol{CMS=\{PA,DA,SA,TA,RA,OA,MA,E\}} \tag{1}$$

where ***PA*** represents the cloud platform agent; ***DA*** represents the cloud demander agent; ***SA*** represents the cloud service agent; ***TA*** represents the cloud task agent; ***RA*** represents the cloud resource agent; ***OA*** represents the order agent issued by ***DA***; ***MA*** represents the message agent sent to ***SA*** when ***TA*** requests or releases resources; and ***E*** represents the external environment of information transmission and inter-entity behavior interaction.

3.1.1. Modeling of Cloud Resource Agent

Through information transformation, resource access, cloud service unified modeling, and other technology, service providers' manufacturing equipment and manufacturing capacity resources are integrated into the cloud resource pool to create virtual resources known as cloud resources. The primary function of the cloud resource agent is to collaborate with cloud tasks to finish the processing of cloud orders:

$$\boldsymbol{RA_0=\langle ID,produceLevel,busy,broken,owner,price\rangle} \tag{2}$$

where ***ID*** represents the resource's special identification number; ***produceLevel*** is an integer ranging from 1 to 10 that specifies the resource's productivity level; ***busy*** indicates whether the resource is in a busy condition; ***broken*** shows whether the resource is defective; ***owner*** identifies which cloud server the resource belongs to; and ***price*** denotes the resource's cost, which is randomly generated using a normal distribution at model startup.

Considering the resource substitution strategies developed in this paper in the face of resource failure (see Section 5.2 for details), it is necessary to continue expanding the attributes of cloud resource agents:

$$\boldsymbol{RA=RA_0+\langle replace\_Resource,replace\_Rate\rangle} \tag{3}$$

where ***replace_Resource*** specifies the alternative resource type R_j for each resource type R_i, with this model assuming that R_i and R_j are mutually substitutive; and ***replace_Rate*** is the replacement rate (i.e., matching rate) of the replacement resource. Although the substitute resource can replace the original resource to complete the established cloud task, there is an increase in the total work time. The resource replacement rate is generated by a normally distributed random number at the time of model initialization.

3.1.2. Modeling of Cloud Task Agent

The development of the cloud task agent is essential to cloud manufacturing simulation modeling. It encompasses both the behavior interaction and information transfer between the cloud server agent and the cloud resource agent, in addition to the processing path for all order kinds (e.g., serial, parallel, and hybrid paths). The cloud task agent further generates (a) the mechanism for choosing the best service provider, (b) numerous statistical data, including the service cycle and cost, and (c) information on the cloud task and cloud resource nodes. The existing process modeling library components are modified accordingly to accomplish this goal. The following is a description of the cloud task agent:

$$\begin{gathered}\boldsymbol{TA_0 = \langle ID, owner\_Orders, pre\_taskList, after\_taskList,}\\ \boldsymbol{reque\_resourceList, basicWorkingTime, currentOrder, Func_{selectBestServer},}\\ \boldsymbol{Func_{selectBestResource}, Func_{recordRouteStamp}, Func_{recordTaskTime}, Func_{recordTaskCost},}\\ \boldsymbol{Func_{recordTaskReliability}\rangle}\end{gathered} \tag{4}$$

where ***ID*** represents the task's special identification number; ***owner_Orders*** identifies which type of order processing path the task belongs to; ***pre_taskList*** and ***after_taskList*** designate the pre-order and post-order tasks, accordingly; ***reque_resourceList*** indicates the resource type requested by the task. ***basicWorkingTime*** specifies the average time required to complete a task; ***currentOrder*** indicates the order being processed at the moment; $\boldsymbol{Func_{selectBestServer}}$ identifies the best server by considering the resource cost, logistics, distance, and other variables; $\boldsymbol{Func_{selectBestResource}}$ decides the best resource; $\boldsymbol{Func_{recordRouteStamp}}$ records the order–task and task–resource relationships for finished orders; $\boldsymbol{Func_{recordTaskTime}}$, $\boldsymbol{Func_{recordTaskCost}}$, and $\boldsymbol{Func_{recordTaskReliability}}$ record, respectively, the service time, the service cost, and the service reliability of the present task.

Again, it is necessary to continue expanding the attributes of cloud task agents:

$$\begin{gathered}\boldsymbol{TA = TA_0 + \langle is\_internalReplace, internal\_replaceResource,}\\ \boldsymbol{internal\_replaceRate, internal\_replaceSerial, is\_externalReplace,}\\ \boldsymbol{external\_Partner, external\_replaceSerial, Func_{calcuWorkingTime}\rangle}\end{gathered} \tag{5}$$

where ***is_internalReplace*** indicates whether the current cloud task process has invoked the internal resource replacement strategy; ***internal_replaceResource*** specifies the replacement resource selected under the internal replacement strategy; ***internal_replaceRate*** is the resource matching rate of the internal replacement resource; ***internal_replaceSerial*** is an integer from 1 to 3 indicating which specific case of internal replacement the current process belongs to; ***is_externalReplace*** indicates whether the current cloud task process has invoked the external resource replacement strategy; ***external_Partner*** indicates the specific external server with which to cooperate under the external resource replacement strategy; ***external_replaceSerial*** is an integer from 1 to 3 indicating which specific case of external replacement the current process belongs to; and $\boldsymbol{Func_{calcuWorkingTime}}$ calculates the cloud task time under the current resource replacement strategy.

Figure 2 depicts the comprehensive CMP simulation implemented within the cloud task agent by modifying and adapting existing component codes from Anylogic's process modeling library. This procedure's specifics are as follows:

(1) By means of the ***enter*** component, the order is imported into the cloud task's internal procedure. The order is immediately assigned by the cloud platform if the current task is the first in the task sequence; if not, the preceding task assigns the order after it has been finished (e.g., task 2 orders are assigned by task 1 after task 1 is finished).
(2) The ***queue*** component temporarily stores the current order while the following determinations are made: (a) if the current task is first in the task sequence or there is only one task in the previous task sequence, the ***hold*** and ***hold1*** components are opened concurrently, and the current order is entered into ***queue2*** for further processing; or (b) if there are multiple tasks in the previous task sequence, the current order must

wait until the orders of all previous tasks have been processed before entering ***queue2*** for further processing.

(3) The ***queue1*** component combines the information of multiple branch orders, whereas the ***hold2*** component ensures that only a single order is entered for subsequent processing at any given time. ***Hold2*** reopens and proceeds to serve the next order when the current order is fulfilled and exits through ***exit***.

(4) When the order enters ***queue3***, the task agent selects the optimal service provider and sends "resource request" information to it. When the optimal service provider accepts the request, it chooses to adopt or not adopt the resource replacement strategy according to whether the target resource is faulty. If the resource substitution strategy is adopted, it is necessary to further select which resource substitution strategy to adopt. The busy attribute of the corresponding optimal resource is changed to "true", the ***hold3*** component opens and the order flows through the ***delay*** component to simulate the cloud manufacturing service. After a certain delay time, the service is completed.

(5) The order is placed in ***queue4*** and the "release resource" message is sent to the best server. When the best server acknowledges the message, the busy attribute of the best resource changes from "true" to "false", and the ***hold4*** component opens. Order traverses the ***delay1*** component, and the release of the resource is accomplished following a predetermined delay period.

(6) The order passes through the **exit** component to conclude all of its service procedures for this task. It then imports the post-order task sequence of this task: (a) if there is only one post-order task, it is imported instantly into the ***enter*** component of the post-order task; (b) if there are numerous post-order tasks, the information of the current order is copied and brought into the ***enter*** component of the corresponding post-order tasks; and (c) if there is no post-order task, this indicates that the task is already the last task in the task sequence. As a result, the order is included in the group of completed orders, and data such as the service cycle, service cost, and route record are tallied and output.

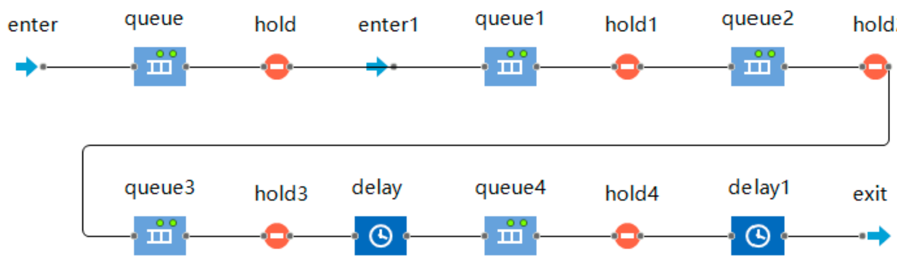


Figure 2. Cloud task agent's internal workflow.

3.1.3. Modeling of Cloud Server Agent

The primary function of cloud servers is to convey information and interact with cloud tasks. Each cloud server's resource pool has all kinds of cloud resources. The server locates the appropriate resource in its resource pool and allots it to the cloud task after receiving the "request resource" message. The server releases the associated cloud resource and adds it back to the cloud resource pool after receiving the "release resource" message. The cloud server agent can be illustrated as follows:

$$\begin{gathered} SA = \langle ID, location, resourcePool, dScore, \\ pScore, totalScore, Func_{configureResource}, \\ Func_{common}, Func_{inter\_replace}, Func_{exter\_replace}, Func_{inter\&exter} \rangle \end{gathered} \quad (6)$$

where ***ID*** represents the cloud service provider's special identification number; ***location*** refers to the server's latitude and longitude coordinates, which is utilized to initialize the server's location on the GIS map; ***resourcePool*** is used for storing the corresponding virtual resources of the cloud server; ***dScore***, ***pScore***, and ***totalScore*** are the distance score, price score, and total score when the cloud task chooses the best cloud server; and $\boldsymbol{Func_{configureResource}}$ is used to manage and distribute resources when receiving cloud task information. If the message is "release resource", the server will locate the corresponding resource and set its busy attribute to "false". If the message is "request resource", the broken attribute of the corresponding resource is examined before any other properties: (1) If the value of the broken property is "false", this indicates that the resource is not faulty, so its busy attribute continues to be judged, where, (a) if the busy attribute is "false", cloud order processing can be started, and (b) if the busy attribute is "true", this means that this resource is being used by other tasks, so it needs to wait until the other tasks have been completed before starting the processing of orders. (2) If the value of the broken property is "true", this means that the resource is faulty and it cannot complete the processing of the corresponding cloud task. If the resource is faulty, according to the experimental settings, one of the following strategies is chosen: the no-resource replacement strategy, the internal resource replacement strategy, the external resource replacement strategy, or the internal–external integrated strategy. $\boldsymbol{Func_{common}}$ is the no-resource replacement strategy: when the requested resource fails, the requested processing order is added to the failed order set. $\boldsymbol{Func_{inter\_replace}}$ is the internal resource replacement strategy: when the requested resource fails, a replacement resource is found for the failed resource within the current service provider. $\boldsymbol{Func_{exter\_replace}}$ is the external resource replacement strategy: when the requested resource fails, the same type of resource as the failed resource from other cloud service providers is requested. $\boldsymbol{Func_{inter\&exter}}$ is a combination strategy of internal and external resource replacement: when the requested resource fails, the priority is to find a replacement resource within the service provider; if the internal replacement resource fails, then external resources are sought from other service providers.

3.1.4. Modeling of Other Agent

In addition to the above types of agents, the CMS also includes multiple types of agents, such as the cloud platform agent, cloud orders agent, cloud messages agent, and cloud demander agent, which can be expressed as:

$$\begin{gathered}\boldsymbol{PA = \langle tobeProcessedOrderList, finishedOrderList,}\\ \boldsymbol{failedOrderList, attackNum, Func_{ini}, Func_{allocationOrders}, Func_{settingFaultStatus},}\\ \boldsymbol{Func_{calcuQos}, Func_{outputNetwork}\rangle}\end{gathered} \tag{7}$$

$$\begin{gathered}\boldsymbol{OA = \langle ID, owner, taskList, routeStamp, cost1Accum, cost2Accum,}\\ \boldsymbol{cost3Accum, reliabilityAccum, startTime, finishTime\rangle}\end{gathered} \tag{8}$$

$$\boldsymbol{MA = \langle msg, resourceList, owner\rangle} \tag{9}$$

$$\boldsymbol{DA = \langle ID, location, orderList, Func_{sendOrders}\rangle} \tag{10}$$

The attributes and methodologies of these four agent types are already detailed in [42], so they are not repeated here.

3.2. Multi-Agent Simulation-Based Robustness Evaluation Indicator

Based on the multi-agent model in Section 3.1 and the order–task sequence, the dynamic simulation of the CMP can be realized, and the results data (e.g., the order completion time, logistics transportation distance, and resource occupation) can be output to evaluate the performance of the CMP. Quality of service (QoS) is commonly used in academia to evaluate the CMP, with QoS values mostly being evaluated from multiple dimensions, such as the time, cost, and reliability [43–45]. Referring to the definition of QoS, this paper proposes robustness of service (RoS) as a robustness measurement indicator from the perspective of the simulation of four dimensions: service time, service cost, reliability, and order completion rate. The specific calculation formulae for these four dimensions are as follows:

(1) Service time

The sum of all orders' completion times during the simulation cycle is represented by the following formula:

$$T = \sum_{j=1}^{m} t_j \tag{11}$$

where m denotes the overall number of orders, $j = 1, 2, \ldots, m$ denotes the jth order in the order sequence, and t_j denotes the finish time of the jth order, which can be derived from the results of the simulation.

(2) Service cost

The cloud resource service fee, logistics service fee, and cloud resource release fee, which together make up the overall cloud service cost, are determined as follows:

$$\begin{aligned} C = \sum_{j=1}^{m} \sum_{i=1}^{n_j} t_{i,j}^{\text{serving}} * p_{i,j}^{\text{resource}} + \sum_{j=1}^{m} \sum_{i=1}^{n_j} d_{i,j} * c^{\text{logistic}} \\ + \sum_{j=1}^{m} \sum_{i=1}^{n_j} t_{i,j}^{\text{releasing}} * p^{\text{release ra}} \end{aligned} \tag{12}$$

where m represents the total amount of orders and n_j represents the number of assignments associated with each order; $j = 1, 2, \ldots, m$ is the jth order in the order sequence; $i = 1, 2, \ldots, n$ is the ith the task in the task sequence; $t_{i,j}^{\text{serving}}$ represents the cloud service time of the ith task in the jth order; $p_{i,j}^{\text{resource}}$ represents the service cost per unit time of the resource related to the task; $d_{i,j}$ represents the logistics distance related to the task; c^{logistic} represents the logistics cost per unit distance; $t_{i,j}^{\text{releasing}}$ represents the release time of the cloud resources for the task; and p^{release} represents the cost per unit time of releasing resources.

(3) Service reliability

The service reliability is measured by a multiplicative index [44], which has the following form:

$$rel = \frac{\sum_{j=1}^{m} \prod_{i=1}^{n_j} rel_{i,j}}{m} \tag{13}$$

where $rel_{i,j}$ represents the service reliability of the ith task in the jth order, as specified in the order agent's $reliabilityAccum$ property.

The evaluation of the CMP's robustness in this paper also refers to the definition of robustness, which refers to the ability of the CMS to operate normally and maintain its original performance in the face of various unexpected disruptions and interruptions. For the CMS, the normal completion of cloud orders within a service cycle can reflect its normal operation capability more intuitively. Therefore, the order completion rate index is introduced:

(4) Order completion rate

The order completion rate is the proportion of total scheduled orders to orders actually completed throughout the simulation cycle.

$$ofr = \frac{N_1}{N_1 + N_2} \tag{14}$$

where N_1 represents the number of orders fulfilled throughout the simulation cycle and N_2 represents the number of incomplete orders.

In summary, combining the definitions of robustness and QoS, this paper proposes a new robustness evaluation index, RoS, to comprehensively consider the normal operational capability of the CMS in the face of unexpected disturbances.

The RoS index takes the order completion rate as the main body, where the higher the order completion rate, the stronger the normal operational ability in the face of interference, and the stronger the ability to resist risks (i.e., the stronger the robustness of the cloud service), and vice versa. When the order completion rates are the same, the index then compares the differences in the time and cost of completing the same order quantity: if the same number of orders are completed with less time, lower cost, and higher reliability, the cloud service is more robust, and vice versa. Therefore, it can be expressed as:

$$RoS = ofr * [1 - (\omega_1 * \frac{|T - T_0|}{T_0} + \omega_2 * \frac{|C - C_0|}{C_0} + \omega_3 * \frac{|rel - rel_0|}{rel_0})] \quad (15)$$

where T_0, C_0, and rel_0 are the respective baseline values of the service time, cost, and reliability under the condition of no interference; T, C, and rel are the actual values of the current experimental group; and ω_1, ω_2, and ω_3 are the respective weight coefficients of the three indexes, satisfying $\sum_{i=1}^{3} \omega_i = 1$.

4. Model of the Cloud Manufacturing Network and Robustness Evaluation Indicator Based on Complex Network

4.1. Development of a Complex Network Model for Cloud Manufacturing

The CMN consists of cloud service resources and their interconnections. The network can be evaluated utilizing the complex network model due to the vast number of resources and intricate connection relationships. Figure 3a depicts the processing task paths for Order-A, Order-B, and Order-C, the resources utilized by each task along these paths, and the relationships between the resources and servers. If two tasks are linked by a path, their corresponding resources are also linked. Figure 3b illustrates how the CMN is formed by considering all resources to be network nodes and resource connections to be connected edges.

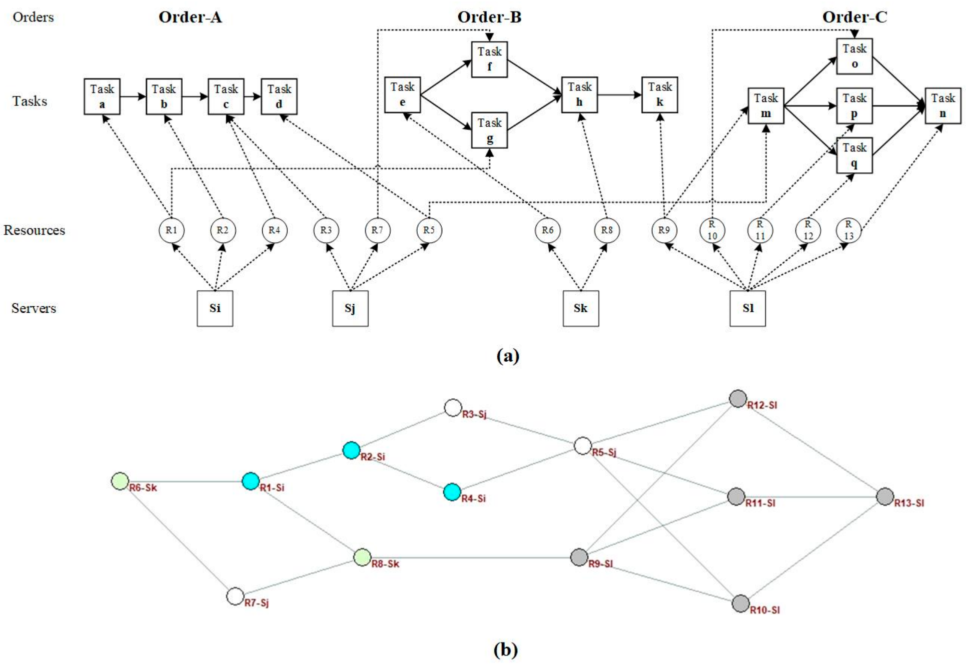


Figure 3. Schematic depiction of (**a**) the CMS's internal relationships and (**b**) the cloud resource network.

4.2. Evaluation Indicator for Network Robustness Based on Static Topology

The term "network robustness" is generally used to describe the degree of network performance retention following the failure of network nodes or edges [44], and the change in the maximum connected subgraph following node failure can reflect the degree of structural integrity retention in the network. As a result, the rate of change in the maximal connected subgraph's node count was chosen as one of the robustness evaluation indicators for this study.

$$S = \frac{N'}{N} \tag{16}$$

where N' represents the number of nodes in the maximally connected subgraph after the network has been attacked, and N represents the number of nodes in the original network. Specifically, $S = 0$ indicates that the network is disconnected, whereas $S = 1$ indicates that the network is completely connected and there are no isolated nodes.

5. Robust Failure Mode Definition and Formulation of Multiple Resource Replacement Strategies

5.1. Definition of Failure Modes for Robustness Analysis

The definition of failure modes is the key to robustness analysis. Based on a combination of the cloud manufacturing characteristics and the spatial topology structure of the CMN, this paper proposes two types of failure modes: cloud resource failure based on the initial topology, and cloud resource failure based on the recomputed topology.

The initial topology refers to the initial structural characteristics of the CMN, which is a static network. The recalculated topology refers to the structural characteristics of the CMN that are obtained through recalculation after the initial network is attacked, which is a dynamic network that changes step by step with the attack steps.

Both failure modes are subdivided into degree-based and betweenness-based resource failures. The degree is widely used to measure the importance of the nodes: it represents how closely a resource node is connected to other resource nodes in the CMN. The betweenness reflects the structural importance of the nodes in the network [46,47]: a node with high betweenness has greater control over the logistics and information flow in the network. The specific failure mode definitions are shown in Table 1.

Table 1. Resource failure based on initial topology and recomputed topology.

	Failure Mode Description	Failure Mode Calculation Process
Resource failure based on initial topology	Initial node degree loss (ID)	Sort the resource nodes in the initial network (Network-0) by degree, from largest to smallest. Remove one node at a time, and repeat n times until all nodes in the network are removed.
	Initial node betweenness loss (IB)	Sort the resource nodes in the initial network (Network-0) by betweenness, from largest to smallest. Remove one node at a time, and repeat n times until all nodes in the network are removed.
Resource failure based on recomputed topology	Recomputed node degree loss (RD)	Sort the resource nodes in the initial network (Network-0) by degree, from largest to smallest. Remove the first node and generate a new network (Network-1). Recalculate and sort the resource nodes in the new network (Network-1) by degree, from largest to smallest. Remove the first node and generate a new network (Network-2) . . . and so on, until all nodes in the network are removed.
	Recomputed node betweenness loss (RB)	Sort the resource nodes in the initial network (Network-0) by betweenness, from largest to smallest. Remove the first node and generate a new network (Network-1). Recalculate and sort the resource nodes in the new network (Network-1) by betweenness, from largest to smallest. Remove the first node and generate a new network (Network-2) . . . and so on, until all nodes in the network are removed.

Note: The removal of nodes is handled differently in the complex network model and the multi-agent model: (1) In the complex network model, the corresponding resource nodes and all connected edges on the nodes are deleted. (2) In the multi-agent model, the corresponding resource agent is changed to a "fault" state, which means the resource is unable to provide services.

5.2. Formulation of Multiple Resource Substitution Strategies

Targeting the cloud resource node failure problem proposed in Section 5.1, this paper aims to develop a robustness enhancement strategy for CMNs that involves adding redundant nodes.

For the supply chain modeling and robustness problem, Zhao [48] took the smartphone supply chain as an example and put forward three different robustness optimization strategies: enterprise internal operation management, cooperative management between enterprises, and a regional development strategy. Inspired by this, and combined with the characteristics of cloud manufacturing, this paper proposes three kinds of robustness improvement strategies: internal resource replacement, external resource replacement, and internal–external integration replacement.

(1) Internal replacement strategy: The cloud service provider *Si* will internally provide replacement resources *Rj* for *Ri* (noted as *Ri-Si* and *Rj-Si*, respectively). If *Ri-Si* fails, *Rj-Si* will replace it to complete the processing of cloud manufacturing tasks. Although the tasks will be completed, the cloud task time will be increased due to the different resource types, and additional working hours will be incurred.

(2) External replacement strategy: The cloud service provider *Sj*, *Sk* ... etc. will provide the same type of resources *Ri* as *Ri-Si* (recorded as *Ri-Sj*, *Ri-Sk* ... etc.). When the resource *Ri* fails, the strategy will comprehensively select the best cloud service provider based on multiple factors, such as resource quotation and the distance between service providers. It will then request alternative resources from them to replace the failed *Ri-Si* to complete the cloud manufacturing task. Since the resource types are the same, this does not add additional task time; however, the transfer of resources and information among different service providers will generate additional logistics transportation costs and time.

(3) Internal–external integration replacement strategy: This strategy is the combination of the previous two strategies. When a cloud resource fails, this strategy first looks for a replacement resource within the service provider; if no replacement resource is found or its replacement resource also fails, it continues to seek the same type of resource from other service providers. The logical flow of these three strategies is shown in Figure 4.

In addition, in the complex network model, node failure is reflected by removing the failed resource node and all the edges connected to the node. After selecting the corresponding resource replacement strategy, the optimal alternative resource node under the current strategy is first determined. If this replacement node is already in the original network, all connected edges belonging to the failed node will be directly linked to the replacement node; if it is not already in the original network, the replacement node should first be added to the network, then be linked similarly.

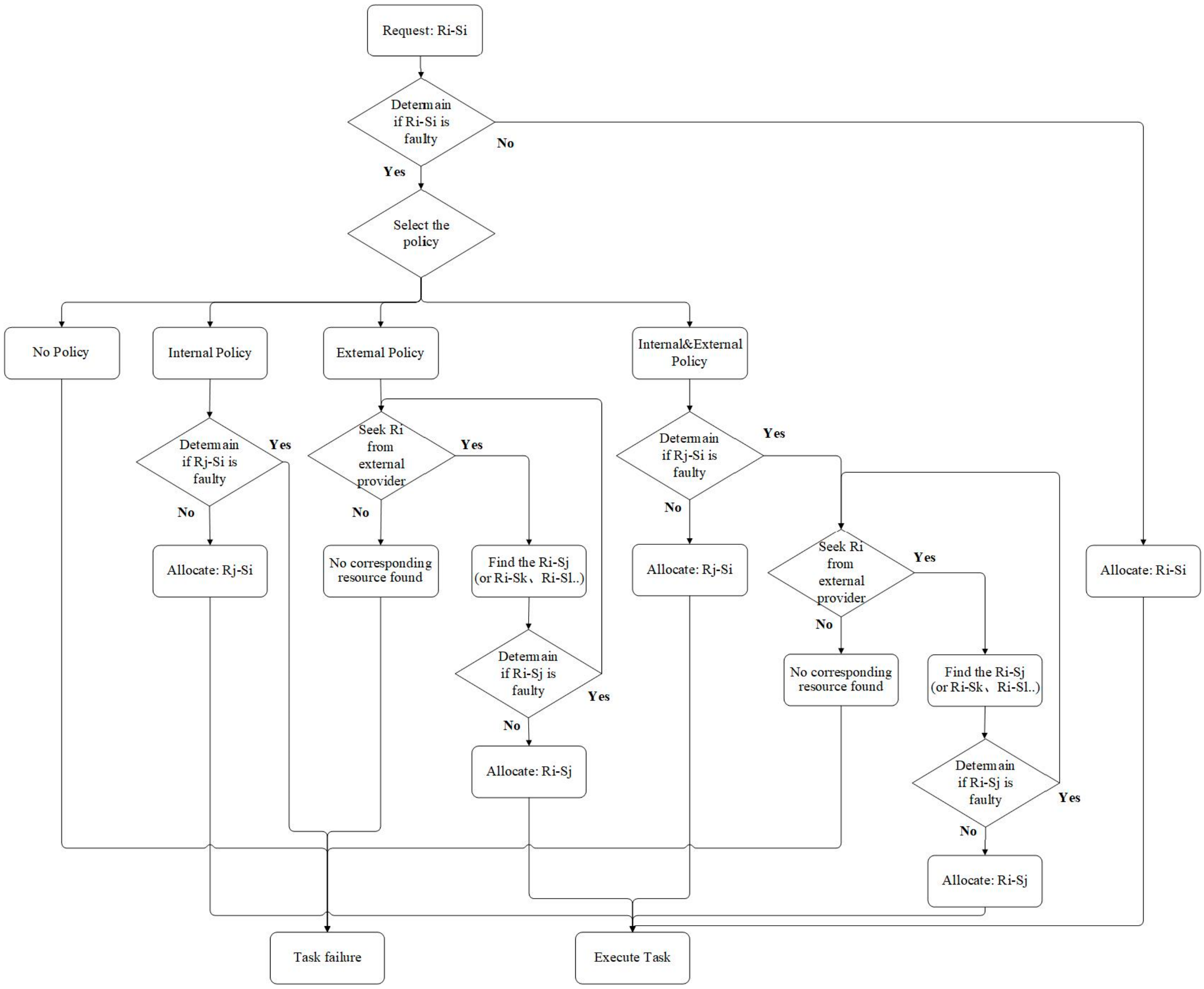


Figure 4. Logical flow diagram of resource substitution strategy.

6. Case Study

6.1. Model Parameters Description

The cloud manufacturing project for a new energy vehicle is used as a case study. This project offers life-cycle cloud manufacturing services for new energy vehicles, with the technologies provided including electrification and autonomous driving.

The cloud manufacturing project consists of 24 order types, 95 cloud tasks (t1–t95), and 72 resource types (r1–r72). Table 2 displays the appropriate resource types for each cloud task, and Table 3 displays the routes for each order type's associated cloud task. This paper assumes a bidirectional substitution relationship between resources (e.g., if the substitute resource of r_i is r_j, the substitute resource of r_j is r_i). Based on this, the substitution relationships among internal resources are shown in Table 4.

Each of the project's five cloud servers (S1-S5) offers 72 different kinds of cloud resources. The cloud servers compete for different orders because they charge different prices for their resources and are located at various distances from cloud demanders. Resource r1 of servers S1–S5 are identified by the labels r1–S1, r1–S2, r1–S3, r1–S4, and r1–S5, respectively.

Moreover, there are 14 cloud demanders (d1–d14). Each cloud demander submits 24 orders, with 1 of each order type submitted (i.e., 1 of each of the 24 order types). As indicated in Table 5, the fundamental details of each cloud service provider and cloud demander are externally imported from Excel.

Table 2. Order and cloud service route correspondence.

Order Type	Cloud Service Route	Order Type	Cloud Service Route
Order 11	t1 → t2	Order 12	t3 → t4
Order 13	t5 → t6	Order 14	t7 → t8
Order 15	t9 → t10	Order 16	t11 → t12 → t13
Order 21	t14 → t15 → t16	Order 22	t17 → t18 → t19
Order 23	t20 → t21 → t22	Order 24	t23 → t24 → t25 → t26
Order 25	t27 → t28 → t29	Order 26	t30 → t31 → t32
Order 31	t33 → t34 → t35 → t36	Order 32	t37 → t38 → t39 → t40
Order 33	t41 → t42 → t44 t41 → t43 → t44	Order 34	t45 → t46 → t48 t45 → t47 → t48
Order 41	t49 → t50 → t51 → t55 t49 → t52 → t53 → t55 t49 → t52 → t54 → t55	Order 42	t56 → t57 → t58 → t62 t56 → t59 → t60 → t62 t56 → t59 → t61 → t62
Order 43	t63 → t64 → t65 → t69 t63 → t66 → t67 → t69 t63 → t66 → t68 → t69	Order 44	t70 → t71 → t72 → t76 t70 → t73 → t74 → t76 t70 → t73 → t75 → t76
Order 51	t77 → t78 → t80 t77 → t79 → t80	Order 52	t81 → t82 → t85 t81 → t83 → t85 t81 → t84 → t85
Order 53	t86 → t87 → t89 t86 → t88 → t89	Order 54	t90 → t91 → t92 → t95 t90 → t93 → t94 → t95

Table 3. Task and resource correspondence.

Task	Resource	Task	Resource	Task	Resource	Task	Resource
t1	(r1)	t2	(r3)	t3	(r2)	t4	(r4)
t5	(r11, r30)	t6	(r5)	t7	(r12, r29)	t8	(r6)
t9	(r12, r29)	t10	(r31)	t11	(r11, r30)	t12	(r67)
t13	(r32)	t14	(r1)	t15	(r7)	t16	(r1)
t17	(r2)	t18	(r8)	t19	(r2)	t20	(r5)
t21	(r5)	t22	(r41)	t23	(r6)	t24	(r6)
t25	(r71)	t26	(r42)	t27	(r33)	t28	(r9)
t29	(r9)	t30	(r34)	t31	(r10)	t32	(r10)
t33	(r9)	t34	(r7)	t35	(r9)	t36	(r7)
t37	(r10)	t38	(r8)	t39	(r10)	t40	(r8)
t41	(r13, r14)	t42	(r61)	t43	(r21, r23)	t44	(r51, r52)
t45	(r15, r16)	t46	(r62)	t47	(r22, r24)	t48	(r53, r54)
t49	(r21, r23)	t50	(r47, r48)	t51	(r47, r48)	t52	(r33)
t53	(r35, r37)	t54	(r63)	t55	(r43, r45)	t56	(r22, r24)

Table 3. *Cont.*

Task	Resource	Task	Resource	Task	Resource	Task	Resource
t57	(r49, r50)	t58	(r49, r50)	t59	(r34)	t60	(r36, r38)
t61	(r64)	t62	(r44, r46)	t63	(r35, r37)	t64	(r41)
t65	(r41)	t66	(r39)	t67	(r25, r27)	t68	(r65)
t69	(r13, r14)	t70	(r36, r38)	t71	(r42)	t72	(r42)
t73	(r40)	t74	(r26, r28)	t75	(r66)	t76	(r15, r16)
t77	(r47, r48)	t78	(r51, r52)	t79	(r57, r58)	t80	(r47, r48)
t81	(r49, r50)	t82	(r53, r54)	t83	(r68)	t84	(r59, r60)
t85	(r49, r50)	t86	(r57, r58)	t87	(r17, r19)	t88	(r63)
t89	(r55)	t90	(r59, r60)	t91	(r18, r20)	t92	(r69)
t93	(r64)	t94	(r70)	t95	(r56)		

Table 4. Substitution relationships among internal resources.

Resource	Alternative Resource	Re-Source	Alternative Resource	Re-Source	Alternative Resource	Re-Source	Alternative Resource
r1	r2	r2	r1	r3	r4	r4	r3
r5	r6	r6	r5	r7	r8	r8	r7
r9	r10	r10	r9	r11	r12	r12	r11
r13	r16	r14	r15	r15	r14	r16	r13
r17	r20	r18	r19	r19	r18	r20	r17
r21	r24	r22	r23	r23	r22	r24	r21
r25	r28	r26	r27	r27	r26	r28	r25
r29	r30	r30	r29	r31	r32	r32	r31
r33	r34	r34	r33	r35	r38	r36	r37
r37	r36	r38	r35	r39	r40	r40	r39
r41	r42	r42	r41	r43	r44	r44	r43
r45	r46	r46	r45	r47	r49	r48	r50
r49	r47	r50	r48	r51	r53	r52	r54
r53	r51	r54	r52	r55	r56	r56	r55
r57	r59	r58	r60	r59	r57	r60	r58
r61	r62	r62	r61	r63	r64	r64	r63
r65	r66	r66	r65	r67	r68	r68	r67
r69	r70	r70	r69	r71	r72	r72	r71

Table 5. Properties of cloud demanders and cloud servers.

ID	City	Location (Latitude, Longitude)	ID	City	Location (Latitude, Longitude)
S1	Beijing	(39.91, 116.41)	d5	Jinan	(36.4, 117)
S2	Shanghai	(31.21, 121.43)	d6	Lanzhou	(36.03, 103.73)
S3	Chengdu	(30.66, 104.06)	d7	Wulumuqi	(43.76, 87.68)
S4	Hangzhou	(30.26, 120.2)	d8	Changsha	(28.21, 113)
S5	Shenzhen	(22.61, 114.06)	d9	Nanchang	(28.68, 115.9)
			d10	Fuzhou	(26.08, 119.3)
d1	HaErbin	(45.75, 126.63)	d11	Nanning	(22.48, 108.19)
d2	ShenYang	(41.8, 123.38)	d12	Lasa	(29.6, 91)
d3	Baotou	(40.39, 109.49)	d13	Lianyungang	(34.36, 119.1)
d4	Tianjin	(39.13, 117.2)	d14	Hefei	(31.52, 117.17)

The weight coefficients for the RoS were set in this study to be $\omega_1 = 1/3$, $\omega_2 = 1/3$, and $\omega_3 = 1/3$.

6.2. Structural Robustness Analysis

Based on the network model construction method described in Section 4.1, Figure 5 depicts the initial cloud manufacturing resource network. Matlab-2020a software was implemented to conduct data statistical analysis on the network, and as shown in Table 6 and Figure 6, the relevant network topology parameters and degree distribution were obtained.

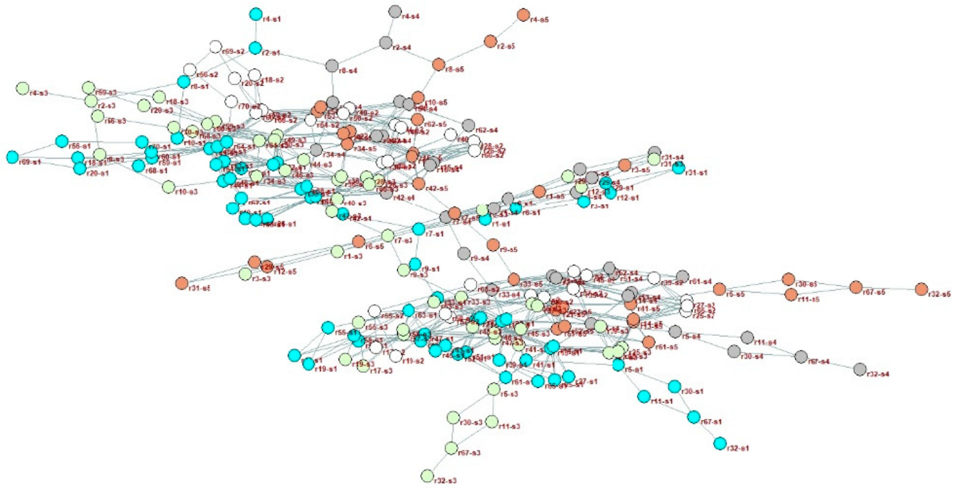


Figure 5. Layout of the cloud resource network in space.

Table 6. Topological parameters of initial cloud resource network.

Topological Parameter	Number of Nodes	Average Degree	Density	Average Path Length
Cloud resource network	231	21.208	0.087	4.231

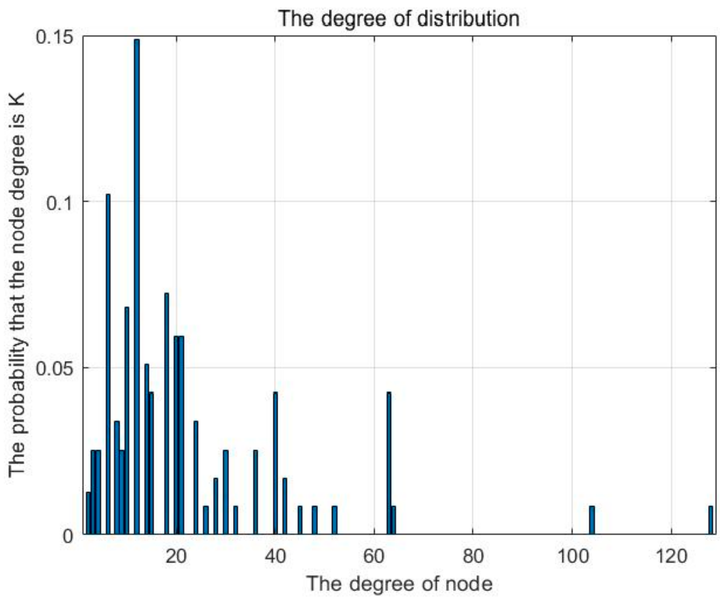


Figure 6. Cloud resource network degree distribution.

The network contained 231 resource nodes, and the distribution of node degrees was very unbalanced. A small number of nodes occupied the vast majority of connected edges, proving that the network possessed the traits of a scale-free network. It was a sparse network, as the nodes with higher degree values tended to connect the nodes with lower degree values, as indicated by the network's low density.

Then, using the four failure modes outlined in Section 5.1, Python 3.0 was used to simulate and determine how the structural robustness indexes changed in response to each failure mode.

6.2.1. Structural Robustness Comparison of Three Substitution Strategies under Initial Node Degree Loss (ID) Failure Mode

As shown in Figure 7, (1) in the ID failure mode, the maximum connectivity subgraphs under the no-substitution strategy ("common") and the three resource substitution strategies ("internal", "external", and "inter&exter") all show a decreasing trend as the number of node failures increases. This indicates that from the complex network perspective, the structural robustness of the CMS gradually decreases. (2) The curves of the three resource substitution strategies are all located above the curve of the no-substitution strategy (i.e., their maximum connectivity subgraph values are larger than the no-substitution strategy's value). This indicates that all three resource substitution strategies can improve the structural robustness of the CMS in the face of initial node degree failure. (3) The "internal" curve is only slightly higher than the "common" curve, and when the number of node attacks is high, the maximum connected subgraph values of these two strategies decrease to 0, at which point the CMN has completely collapsed. In contrast, the "external" and "inter&exter" curves are significantly higher than the "common" curve, and even when the number of node attacks is high, the connected subgraph values still maintain a high level. This indicates that the external replacement strategy and the internal–external integrated replacement strategy both offer more significant robustness enhancement than the internal replacement strategy under the ID failure mode.

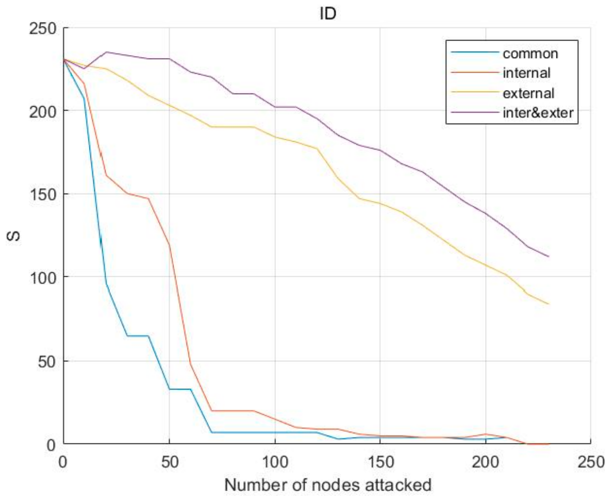


Figure 7. Variation of S under ID failure mode.

6.2.2. Structural Robustness Comparison of Three Substitution Strategies under Initial Node Betweenness Loss (IB) Failure Mode

As shown in Figure 8, (1) in the IB failure mode, the maximum connectivity subgraphs under the no-substitution strategy ("common") and the three resource substitution strategies ("internal", "external", and "inter&exter") all show a decreasing trend as the number of node failures increases. This indicates that from the complex network perspective, the structural robustness of the CMS gradually decreases. (2) The curves of the three resource substitution strategies are all located above the curve of the no-substitution strategy (i.e., their maximum connectivity subgraph values are larger than the no-substitution strategy's value). This indicates that all three resource substitution strategies can improve the structural robustness of the CMS in the face of initial node betweenness failure. (3) The "internal" curve is only slightly higher than the "common" curve, and when the number of node attacks is high, the maximum connected subgraph values of these two strategies decrease to 0, at which point the CMN has completely collapsed. In contrast, the "external" and "inter&exter" curves are significantly higher than the "common" curve, and even when the number of node attacks is high, the connected subgraph values still maintain a high level. This indicates that the external replacement strategy and the internal–external integrated replacement strategy both offer more significant robustness enhancement than the internal replacement strategy under the IB failure mode.

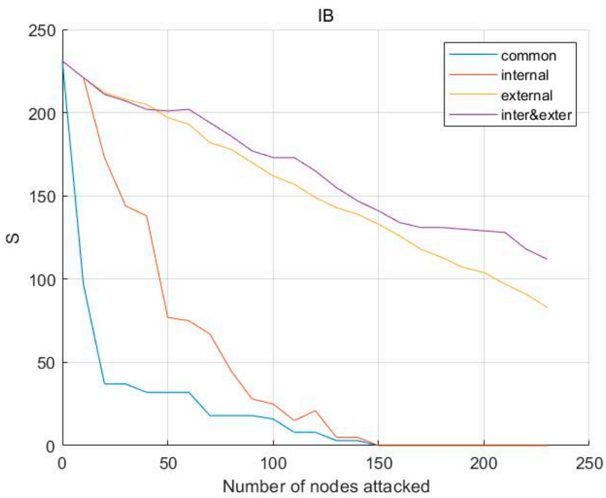


Figure 8. Variation of S under IB failure mode.

6.2.3. Structural Robustness Comparison of Three Substitution Strategies under Recomputed Node Degree Loss (RD) Failure Mode

As shown in Figure 9, (1) in the RD failure mode, when the number of node attacks is high, the maximum connectivity subgraph values under all three substitution strategies decrease to 0, at which point the CMN has completely collapsed. This indicates that the RD failure mode is more destructive to the structural robustness of the CMS than either the ID or IB modes. (2) The "external" curve is always located above the "internal" curve, and the "inter&exter" curve is nearly always located above the "external" curve. This indicates that the maximum connectivity subgraph value is largest under the internal–external integration replacement strategy, followed by the external replacement strategy, and then the internal replacement strategy. Therefore, for the structural robustness of the CMS under the RD failure mode: internal–external integration replacement strategy > external replacement strategy > internal replacement strategy.

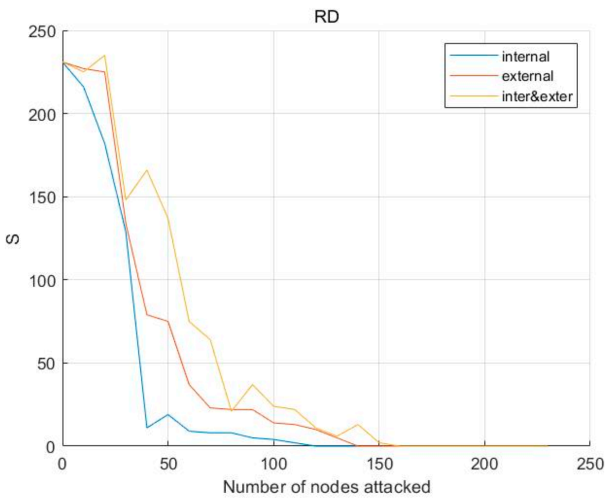


Figure 9. Variation of S under RD failure mode.

6.2.4. Structural Robustness Comparison of Three Substitution Strategies under Recomputed Node Betweenness Loss (RB) Failure Mode

As shown in Figure 10, (1) in the RB failure mode, when the number of node attacks is high, the maximum connectivity subgraph values under all three strategies decrease to 0, at which point the CMN has completely collapsed. This indicates that the RB failure mode is more destructive to the structural robustness of the CMS than either the ID or IB modes. (2) The "external" curve is always located above the "internal" curve, and the majority of the "inter&exter" curve is located above the "external" curve. This indicates that the maximum connectivity subgraph value is largest under the internal–external integration replacement strategy, followed by the external replacement strategy, and then the internal replacement strategy. Therefore, for the structural robustness of the CMS under the RB failure mode: internal–external integration replacement strategy > external replacement strategy > internal replacement strategy.

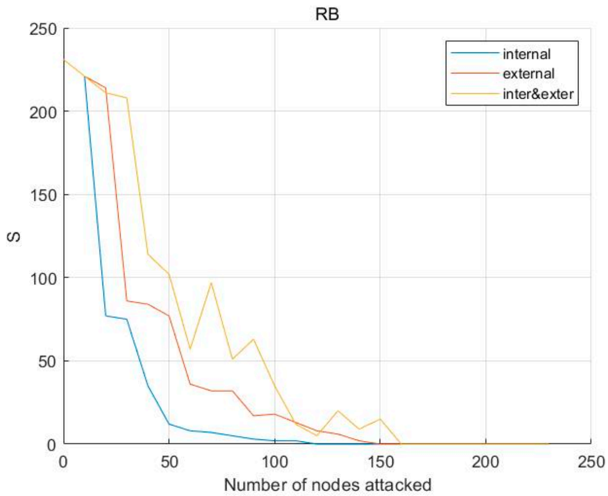


Figure 10. Variation of S under RB failure mode.

In summary, from the complex network perspective, all three resource substitution strategies significantly improved the structural robustness of the CMS. In the four failure modes (i.e., ID, IB, RD, and RB), the structural robustness levels under all three strategies were higher than those with no strategy. Further, the internal–external integration replacement strategy brought the greatest robustness enhancement to the CMS, followed by the external replacement strategy, and then the internal replacement strategy. This is reasonable because the essence of a resource substitution strategy is to add redundant nodes, so when a resource fails, redundant alternative resources will be there to replace it to complete the task. Therefore, as the number of node failures increased, the strategies with more initial redundant nodes (i.e., the internal–external resource integration strategy and the external replacement strategy) were more robust. Conversely, the strategy with fewer initial redundant nodes (i.e., the internal replacement strategy) was less robust. In addition, in the failure modes based on the initial topology (i.e., ID and IB), only the structural robustness under the no-replacement strategy and the internal replacement strategy significantly decreased. In contrast, in the failure modes based on the recomputed topology (i.e., RD and RB), the structural robustness under all strategies significantly decreased, indicating that the failure modes based on the recomputed topology were more destructive to the structural robustness of the CMS. However, for all four failure modes, all three resource substitution strategies could protect the structural robustness of the CMS to some extent.

6.3. Process Robustness Analysis

This research analyzed changes in the RoS under the four failure types using the multi-agent simulation program Anylogic and Python 3.0.

6.3.1. Process Robustness Comparison of Three Substitution Strategies under ID Failure Mode

As shown in Figure 11, (1) in the ID failure mode, the RoS values under the no-substitution strategy ("common") and the three resource substitution strategies ("internal", "external", and "inter&exter") all show a decreasing trend as the number of node failures increases (i.e., as the cloud order completion rate decreases). This indicates that from a multi-agent simulation perspective, the robustness of the CMP gradually decreases. (2) The curves of the three resource substitution strategies are all located above the curve of the no-substitution strategy (i.e., their RoS values are larger than the no-substitution strategy's value, which means their order completion rates are higher). This indicates that all three resource substitution strategies can improve the robustness of the CMP in the face of initial node degree failure. (3) The "internal" curve is only slightly higher than the "common" curve, and when the number of node attacks is high, the RoS values of these two strategies decrease to 0, at which point all cloud orders fail to be processed. In contrast, the "external" and "inter&exter" curves are significantly higher than the "common" curve, and though the RoS values fluctuate when the number of node attacks is high, they still maintain a high level (above 0.6). This indicates that the external replacement strategy and the internal–external integration replacement strategy both offer more significant robustness enhancement than the initial replacement strategy under the ID failure mode. In particular, when the number of node attacks is high, the slight fluctuations of the RoS value indicate that the cloud order completion rate tends to be stable at this time, but different resource substitution strategies will lead to changes in the service time, cost, reliability, and other factors.

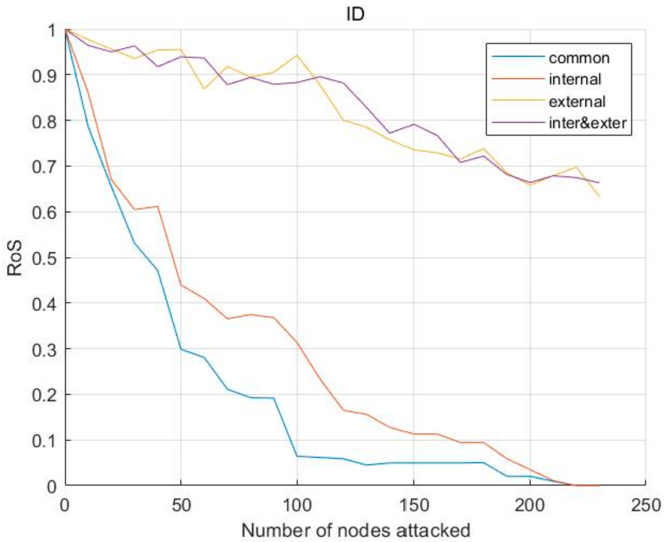


Figure 11. Variation of RoS under ID failure mode.

6.3.2. Process Robustness Comparison of Three Substitution Strategies under IB Failure Mode

As shown in Figure 12, (1) in the IB failure mode, the RoS values under the no-substitution strategy ("common") and the three resource substitution strategies ("internal", "external", and "inter&exter") all show a decreasing trend as the number of node failures increases (i.e., as the cloud order completion rate decreases). This indicates that from a multi-agent simulation perspective, the robustness of the CMP gradually decreases. (2) The curves of the three resource substitution strategies are all located above the curve of the no-substitution strategy (i.e., their RoS values are larger than the no-substitution strategy's value, which means their order completion rates are higher). This indicates that all three resource substitution strategies can improve the robustness of the CMP in the face of initial node betweenness failure. (3) The "internal" curve is only slightly higher than the "common" curve, and when the number of node attacks is high, the RoS values of these two strategies decrease to 0, at which point all cloud orders fail to be processed. In contrast, the "external" and "inter&exter" curves are significantly higher than the "common" curve, and though the RoS values fluctuate when the number of node attacks is high, they still maintain a high level (above 0.6). This indicates that the external replacement strategy and the internal–external integration replacement strategy both offer more significant robustness enhancement than the initial replacement strategy under the IB failure mode. In particular, when the number of node attacks is high, the slight fluctuations in the RoS value indicate that the cloud order completion rate tends to be stable at this time, but different resource substitution strategies will lead to changes in the service time, cost, reliability, and other factors.

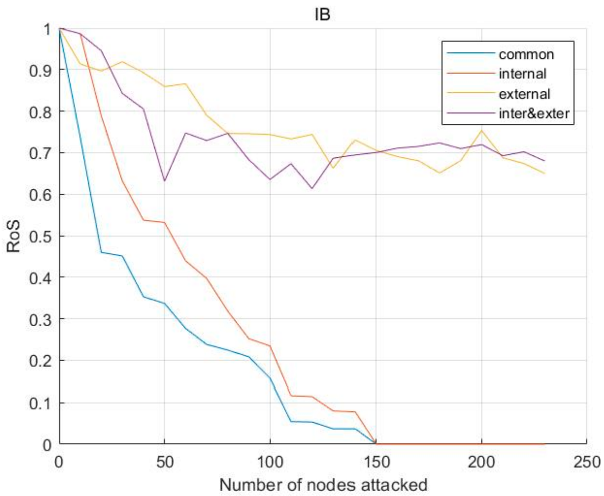


Figure 12. Variation of RoS under IB failure mode.

6.3.3. Process Robustness Comparison of Three Substitution Strategies under RD Failure Mode

As shown in Figure 13, (1) in the RD failure mode, as the number of node attacks increases, the RoS values under all three substitution strategies rapidly decline and finally decrease to 0, at which point all cloud orders fail to be processed. This indicates that the RD failure mode is more destructive to the robustness of the CMP than either the ID or IB modes. (2) The "external" curve is always located above the "internal" curve, and the "inter&exter" curve is nearly always located above the "external" curve. This indicates that the RoS value is largest under the internal–external integration replacement strategy, followed by the external substitution strategy, and then the internal substitution strategy. Further, under the three strategies, the numbers of node attacks required to make all cloud order processing fail (i.e., when the process robustness decreases to its lowest) are approximately 110 ("internal"), 140 ("external"), and 160 ("inter&exter"). Therefore, to attain robustness of the CMP under the RD failure mode: internal–external integration strategy > external substitution strategy > internal substitution strategy.

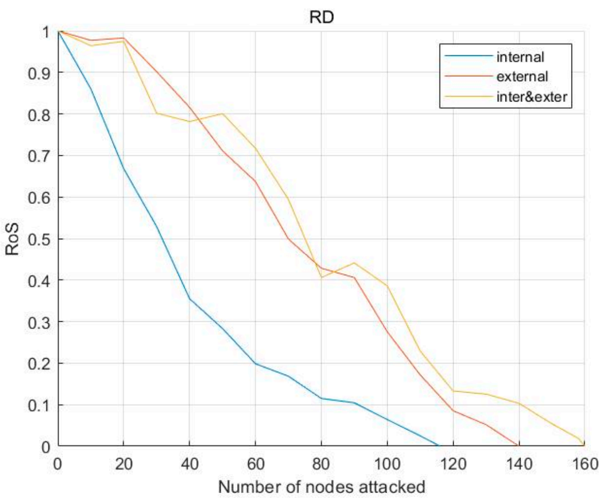


Figure 13. Variation of RoS under RD failure mode.

6.3.4. Process Robustness Comparison of Three Substitution Strategies under RB Failure Mode

As shown in Figure 14, (1) in the RB failure mode, as the number of node attacks increases, the RoS values under all three substitution strategies rapidly decline and finally decrease to 0, at which point all cloud orders fail to be processed. This indicates that the RB failure mode is more destructive to the robustness of the CMP than either the ID or IB modes. (2) The "external" curve is always located above the "internal" curve, and the "inter&exter" curve is nearly always located above the "external" curve. This indicates that the RoS value is largest under the internal–external integration replacement strategy, followed by the external substitution strategy, and then the internal substitution strategy. Further, under the three strategies, the number of node attacks required to make all cloud order processing fail (i.e., when the process robustness decreases to its lowest) are 110 ("internal"), 150 ("external"), and 160 ("inter&exter"). Therefore, to obtain robustness of the CMP under the RB failure mode: internal–external integration strategy > external substitution strategy > internal substitution strategy.

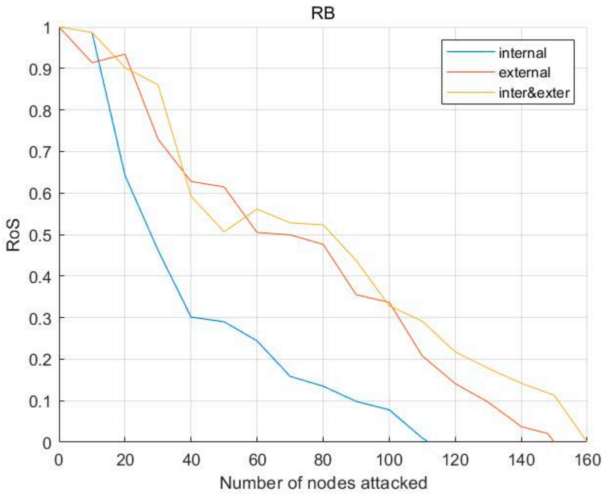


Figure 14. Variation of RoS under RB failure mode.

In summary, from the perspective of multi-agent simulation, all three resource substitution strategies significantly improved the process robustness of the CMS. In the four failure modes (i.e., ID, IB, RD, and RB), the process robustness levels under all three strategies were higher than those with no strategy. Further, the internal–external integration replacement strategy brought the greatest robustness enhancement, followed by the external replacement strategy, and then the internal replacement strategy. This is reasonable because the three strategies provided different amounts of alternative resources: the internal substitution strategy can provide fewer alternative resources, which corresponds to lower robustness; the external substitution strategy can provide five resources of the same type because five external cloud service providers are involved in this paper; and the internal–external integration replacement strategy can provide more than five resources of the same type or alternative resources, which corresponds to higher robustness. In addition, in the failure modes based on the initial topology (i.e., ID and IB), only the process robustness under the no-replacement strategy and the internal replacement strategy significantly decreased. In contrast, in the failure modes based on the recomputed topology (i.e., RD and RB), the process robustness under all strategies significantly decreased, indicating that the failure modes based on the recomputed topology were more destructive to process robustness. However, for all four failure modes, all three resource substitution strategies could protect the process robustness of the CMS to some extent.

6.4. Management Suggestion

Based on the analysis results in Sections 6.2 and 6.3, the following management suggestions were obtained. First, focus should be placed on protecting the resource nodes with a larger degree and larger betweenness (i.e., the important nodes in the CMS). The structural robustness and process robustness of the CMS decreased rapidly in the failure modes based on the node degree (i.e., ID and RD) and node betweenness (i.e., IB and RB), indicating that nodes with a larger degree and betweenness are crucial to maintaining system robustness. More specifically, nodes with a large degree are closely connected with other nodes, so they can play an important role in maintaining system connectivity, and nodes with a large betweenness have greater control over the logistics and information flow in the system, so they can play an important role in maintaining the information transmission rate of the system. Second, alternative resources must be provided to ensure that when the original resources fail, alternative resources can replace them to complete their tasks. These alternative resources can be set up within (1) the same service provider, (2) other external service providers, or (3) a combination of both, and all these methods can protect the robustness of the CMS to a certain extent.

7. Conclusions

This study combined the complex network with multi-agent simulation to propose a new analysis method for the structural robustness and process robustness of the CMS. To target the frequent failure of resource nodes in the cloud manufacturing environment, three resource substitution strategies were proposed to better ensure the stability and robustness of the system. First, a multi-agent simulation model was constructed to study the dynamic process robustness of the CMS. Here, RoS was proposed as a robustness measure, and the behavior characteristics and modeling methods of several key types of CMP agents were detailed. Second, a complex network model of cloud manufacturing resources was established through the order–task relationship and task–resource relationship to study the static topological robustness of the CMS. Here, the maximum connectivity subgraph was proposed as a robustness measure. Regarding attack strategies, four failure modes (i.e., ID, IB, RD, and RB) were defined, and regarding robustness enhancement strategies, three resource substitution strategies (i.e., internal replacement, external replacement, and internal–external integration replacement) were proposed. Third, a case study of a cloud manufacturing project of a new energy vehicle was conducted. The results of this show that (1) the proposed RoS index was effective at portraying the variations of CMP robustness, (2) the three resource substitution strategies could improve both the structural robustness and process robustness of the CMS (with the internal–external integration strategy being most effective, followed by the external substitution strategy, and then the internal substitution strategy), and (3) the two node failure modes based on the recalculated topology were more destructive to the robustness of the CMP than the two node failure modes based on the initial topology. However, for all four failure modes, all three resource substitution strategies could protect the robustness of the CMS to some degree.

In combining the complex network with multi-agent simulation, the robustness analysis object of the CMS is extended from the CMN to the CMP, which provides a new perspective with two dimensions (i.e., structure and process). Moreover, the three proposed recovery strategies (elastic measures) are designed based on the idea of adding redundant nodes, which is of great significance to the implementation and deployment of cloud manufacturing projects. This research will be furthered by investigating the robustness of cloud path interruption, cloud logistics interruption, city lockdowns, and other phenomena, to provide a quantitative and dynamic decision-making basis for improving the robustness of the CMS.

Author Contributions: Conceptualization, X.Z. (Xin Zheng); methodology, X.Z. (Xiaodong Zhang).; software, X.Z. (Xin Zheng); formal analysis, X.Z. (Xin Zheng); investigation, X.Z. (Xin Zheng); resources, X.Z. (Xin Zheng); data curation, X.Z. (Xin Zheng) and Y.W.; writing—original draft

preparation, X.Z. (Xin Zheng); writing—review and editing, X.Z. (Xin Zheng) and X.Z. (Xiaodong Zhang); visualization, X.Z. (Xin Zheng) and Y.W.; supervision, X.Z. (Xiaodong Zhang); project administration, X.Z. (Xiaodong Zhang). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 71871018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The used and analyzed datasets during the present study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Li, B.; Chai, X.; Hou, B.; Li, T.; Zhang, Y.; Yu, H.; Han, J.; Di, Y.; Huang, J.; Song, C.; et al. Networked Modeling & Simulation Platform Based on Concept of Cloud Computing—Cloud Simulation Platform. *J. Syst. Simul.* **2009**, *21*, 5292–5299.
2. Li, B.; Zhang, L.; Wang, S.; Tao, F.; Cao, J.; Jiang, X.; Song, X.; Chai, X. Cloud manufacturing: A new service-oriented networked manufacturing model. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 1–7.
3. Zhao, C.; Zhang, L.; Liu, Y.; Zhang, Z.; Yang, G.; Li, B.H. Agent-based simulation platform for cloud manufacturing. *Int. J. Model. Simul. Sci. Comput.* **2017**, *8*, 1742001. [CrossRef]
4. Jacob, I.; Lu, Y.; Xun, X. Cloud Manufacturing–An Overview of Developments In Critical Areas, Prototypes, and Future Perspectives. *IFAC-Pap.* **2022**, *55*, 643–648. [CrossRef]
5. Narayan Rane, T. System and Risk Analysis of Cloud-Manufacturing System. *Int. J. Comput. Sci. Eng. Surv.* **2022**, *13*, 13–27. [CrossRef]
6. Zeng, Z.; Wu, Q.; Yan, Y. Analysis of Influencing Factors on Cloud Service Composition Flexibility: Based on Perspective of Cloud Manufacturing Service Platform. *Sci. Technol. Manag. Res.* **2019**, *39*, 234–239.
7. Zhang, Y.; Zhang, L.; Liu, Y.; Luo, X. Proof of service power: A blockchain consensus for cloud manufacturing. *J. Manuf. Syst.* **2021**, *59*, 1–11. [CrossRef]
8. Zhu, X.; Shi, J.; Huang, S.; Zhang, B. Consensus-oriented cloud manufacturing based on blockchain technology: An exploratory study. *Pervasive Mob. Comput.* **2020**, *62*, 101113. [CrossRef]
9. Laili, Y.; Lin, S.; Tang, D. Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment. *Robot Comput.-Int. Manuf.* **2020**, *61*, 101850. [CrossRef]
10. Wang, Y.; Wang, S.; Kang, L.; Wang, S. An effective dynamic service composition reconfiguration approach when service exceptions occur in real-life cloud manufacturing. *Robot Comput. Int. Manuf.* **2021**, *71*, 102143. [CrossRef]
11. Liang, H.; Wen, X.; Liu, Y.; Zhang, H.; Zhang, L.; Wang, L. Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. *Robot Comput. Int. Manuf.* **2021**, *67*, 101991. [CrossRef]
12. Tao, F.; Zhang, L.; Guo, H.; Luo, Y.; Ren, L. Typical characteristics of cloud manufacturing and several key issues of cloud service composition. *Comput. Integr. Manuf. Syst.* **2011**, *17*, 477–486.
13. Gao, Z.; Zhu, Q.B.; Song, T.X. In Analysis and research on dynamic models of complex manufacturing network cascading failures. In Proceedings of the 2014 Sixth International Conference on Intelligent Human-Machine Systems AND Cybernetics (IHMSC), Hangzhou, China, 26–27 August 2014; Volume 1, pp. 388–391.
14. Su, Y.; Chai, G.; Liu, P. Topological Structure and Vulnerability of Cloud Manufacturing Network. In Proceedings of the International Conference on Artificial Intelligence and Communication Technology (AICT 2020), Chongqing, China, 7–9 October 2020; p. 5.
15. Cauvin, A.C.A.; Ferrarini, A.F.A.; Tranvouez, E.T.E. Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours. *Int. J. Prod. Econ.* **2009**, *122*, 429–439. [CrossRef]
16. Li, P.; Cheng, Y.; Tao, F. Failures detection and cascading analysis of manufacturing services collaboration toward industrial internet platforms. *J. Manuf. Syst.* **2020**, *57*, 169–181. [CrossRef]
17. Lachenmaier, J.F.; Lasi, H.; Kemper, H. Simulation of Production Processes Involving Cyber-physical Systems. *Procedia CIRP* **2017**, *62*, 577–582. [CrossRef]
18. Galaske, N.; Anderl, R. Disruption Management for Resilient Processes in Cyber-physical Production Systems. *Procedia CIRP* **2016**, *50*, 442–447. [CrossRef]
19. Hou, Y. Study on Robust Scheduling of Flexible Job Shop based on Process Relationship Network under Machine Failure. Master's Thesis, Xinjiang University, Urumqi, China, 2020.
20. Zhang, Z. Research on Robustness of Manufacturing Product Warranty Network. Master's Thesis, Hangzhou Dianzi University, Hangzhou, China, 2012.

21. Beygelzimer, A.; Grinstein, G.; Linsker, R.; Rish, I. Improving network robustness by edge modification. *Phys. A Stat. Mech. Its Appl.* **2005**, *357*, 593–612. [CrossRef]
22. Ma, C. Research on Robustness Evaluation of Urban Gas Pipeline Network based on Cascade Failure. Master's Thesis, North China University of Technology, Beijing, China, 2022.
23. Ji, X.; Wang, B.; Liu, D.; Chen, G.; Tang, F.; Wei, D.; Tu, L. Improving interdependent networks robustness by adding connectivity links. *Phys. A Stat. Mech. Its Appl.* **2016**, *444*, 9–19. [CrossRef]
24. Li, X. Study on the Robustness of Complex Production Process under Disturbing Environment. Master's Thesis, Xinjiang University, Urumqi, China, 2018.
25. Wang, S. Robust Analysis of Production Supply Chain Network. Master's Thesis, Hangzhou Dianzi University, Hangzhou, China, 2021.
26. Wang, X.; Cao, J.; Li, R.; Zhao, T. A preferential attachment strategy for connectivity link addition strategy in improving the robustness of interdependent networks. *Phys. A Stat. Mech. Its Appl.* **2017**, *483*, 412–422. [CrossRef]
27. Chen, C.; Zhao, Y.; Qin, H.; Meng, X.; Gao, J. Robustness of interdependent scale-free networks based on link addition strategies. *Phys. A Stat. Mech. Its Appl.* **2022**, *604*, 127851. [CrossRef]
28. Kazawa, Y.; Tsugawa, S. Effectiveness of link-addition strategies for improving the robustness of both multiplex and interdependent networks. *Phys. A Stat. Mech. Its Appl.* **2020**, *545*, 123586. [CrossRef]
29. Bachmann, I.; Valdés, V.; Bustos-Jiménez, J.; Bustos, B. Effect of adding physical links on the robustness of the Internet modeled as a physical–logical interdependent network using simple strategies. *Int. J. Crit. Infr. Prot.* **2022**, *36*, 100483. [CrossRef]
30. Cao, X.; Hong, C.; Du, W.; Zhang, J. Improving the network robustness against cascading failures by adding links. *Chaos Solitons Fractals* **2013**, *57*, 35–40. [CrossRef]
31. Zhu, L.; Li, P.; Yang, X.; Shen, G.; Zhao, Y. EE-RJMTFN: A novel manufacturing risk evaluation method for alternative resource selection in cloud manufacturing. *Concurr. Eng.* **2019**, *27*, 3–13. [CrossRef]
32. Barbosa, J.; Leitão, P.; Adam, E.; Trentesaux, D. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Comput. Ind.* **2015**, *66*, 99–111. [CrossRef]
33. Dobrescu, R.; Merezeanu, D.; Mocanu, S. Process simulation platform for virtual manufacturing systems evaluation. *Comput. Ind.* **2019**, *104*, 131–140. [CrossRef]
34. Chen, T.; Chiu, M. Development of a cloud-based factory simulation system for enabling ubiquitous factory simulation. *Robot Comput. Int. Manuf.* **2017**, *45*, 133–143. [CrossRef]
35. Zhang, L.; Zhou, L.; Ren, L.; Laili, Y. Modeling and simulation in intelligent manufacturing. *Comput. Ind.* **2019**, *112*, 103123. [CrossRef]
36. Zhao, C.; Luo, X.; Zhang, L. Modeling of service agents for simulation in cloud manufacturing. *Robot Comput. Int. Manuf.* **2020**, *64*, 101910. [CrossRef]
37. Zhao, C.; Wang, L.; Zhang, X. Service agent networks in cloud manufacturing: Modeling and evaluation based on set-pair analysis. *Robot Comput. Int. Manuf.* **2020**, *65*, 101970. [CrossRef]
38. Eisa, M.; Younas, M.; Basu, K.; Awan, I. Modelling and Simulation of QoS-Aware Service Selection in Cloud Computing. *Simul. Model Pract. Theory* **2020**, *103*, 102108. [CrossRef]
39. Zhou, L.; Zhang, L.; Ren, L. Modelling and simulation of logistics service selection in cloud manufacturing. *Procedia CIRP* **2018**, *72*, 916–921. [CrossRef]
40. Vespoli, S.; Grassi, A.; Guizzi, G.; Santillo, L.C. Evaluating the advantages of a novel decentralised scheduling approach in the Industry 4.0 and Cloud Manufacturing era. *IFAC Pap.* **2019**, *52*, 2170–2176. [CrossRef]
41. Li, B.; Zhang, L.; Ren, L.; Chai, X.; Tao, F.; Luo, Y.; Wang, Y.; Yin, C.; Huang, G.; Zhao, X. Further discussion on cloud manufacturing. *Comput. Integr. Manuf. Syst.* **2011**, *17*, 449–457.
42. Zheng, X.; Zhang, X. Robustness of Cloud Manufacturing System Based on Complex Network and Multi-Agent Simulation. *Entropy* **2023**, *25*, 45. [CrossRef] [PubMed]
43. Wang, Y.; Wang, S.; Gao, S.; Guo, X.; Yang, B. Adaptive multi-objective service composition reconfiguration approach considering dynamic practical constraints in cloud manufacturing. *Knowl. Based Syst.* **2021**, *234*, 107607. [CrossRef]
44. Yang, B.; Wang, S.; Cheng, Q.; Jin, T. Scheduling of field service resources in cloud manufacturing based on multi-population competitive-cooperative GWO. *Comput. Ind. Eng.* **2021**, *154*, 107104. [CrossRef]
45. Huang, B.; Li, C.; Tao, F. A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system. *Enterp. Inf. Syst.* **2014**, *8*, 445–463. [CrossRef]
46. Zhou, H. Research on Dynamic Robustness of Knowledge Collaboration Network of Open Source Product Community. Ph.D. Thesis, University of Science and Technology Beijing, Beijing, China, 2018.
47. Lei, S. Robustness of Open-Source Community Knowledge Collaboration Network based on Multiple Failure Modes. Ph.D. Thesis, University of Science and Technology Beijing, Beijing, China, 2022.
48. Zhao, Q. Supply Chain Modeling and Robustness Analysis in the Perspective of Complex Network-a Case Study of Smartphone Supply Chain. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2021.

applied sciences

MDPI

Article

Load Balancing of Two-Sided Assembly Line Based on Deep Reinforcement Learning

Guangpeng Jia [1], Yahui Zhang [2,*], Shuqi Shen [1], Bozu Liu [1], Xiaofeng Hu [3] and Chuanxun Wu [2]

[1] Process Research Institution, China National Heavy Duty Truck Group Co., Ltd., Jinan 250100, China
[2] 5G Intelligent Manufacturing Research Center, Institute of Marine Equipment, Shanghai Jiao Tong University, Shanghai 200240, China
[3] Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
* Correspondence: emily187@sjtu.edu.cn

Abstract: In the complex and ever-changing manufacturing environment, maintaining the long-term steady and efficient work of the assembly line is the ultimate goal pursued by relevant enterprises, the foundation of which is a balanced load. Therefore, this paper carries out research on the two-sided assembly line balance problem (TALBP) for load balancing. At first, a mathematical programming model is established with the objectives of optimizing the line efficiency, smoothness index, and completion time smoothness index of the two-sided assembly line (TAL). Secondly, a deep reinforcement learning algorithm combining distributed proximal policy optimization (DPPO) and the convolutional neural network (CNN) is proposed. Based on the distributed reinforcement learning agent structure assisted by the marker layer, the task assignment states of the two-sided assembly and decisions of selecting tasks are defined. Task assignment logic and reward function are designed according to the optimization objectives to guide task selection and assignment. Finally, the performance of the proposed algorithm is verified on the benchmark problem.

Keywords: two-sided assembly line; load balancing; deep reinforcement learning; distributed multiple processes

1. Introduction

An assembly line (AL) is an arrangement of workstations in a streamlined manner according to the product assembly process sequence for the organization and the arrangement of production. A workstation is an assembly unit that focuses on a specific segment of production, and the workpiece is assembled in all the workstations to form a complete product. Due to the assembly line adopting the flow operation mode, the assembly operation process is standardized, and the assembly workers are generally fixed in a single station or several adjacent stations for repeated operations, which increases the utilization rate of workers, thus greatly improving production efficiency [1].

In the two-sided assembly line (TAL), the left and right sides of the same workstation can independently execute assembly of the same product with different processes in parallel, as shown in Figure 1. These are called mated stations [2]. Compared with the one-sided AL, the TAL can effectively shorten the length of theAL, improve the utilization rate of auxiliary tools, reduce the time loss caused by the movement of workers between various stations, and reduce the transportation cost of assembly parts [2].

To balance the TAL is to distribute a group of tasks evenly to each station as far as possible under certain constraints to pursue the optimization of one or more objectives [3]. However, after the assembly line is put into operation, the original balance parameters, such as cycle time, operation content, operation time, and assembly process, may change with the improvement of assembly workers' technology, product upgrades, customer demand change, etc. In such cases, the original assembly line balance may be disrupted, prohibiting the assembly line from being a stable, efficient, and high-quality operation, and thus greatly reducing the economic benefits of the relevant enterprises. Therefore, it is necessary to optimize and improve the original balancing scheme, which involves reassigning tasks to achieve the assembly line load balance and improve the operation efficiency of the TAL.

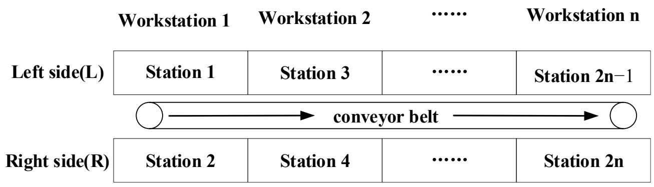


Figure 1. Two-sided assembly line.

TALBP is one of the NP-hard problems belonging to combinatorial optimization [4]. Since it was proposed in 1993 [5], it has been widely studied. The main research methods include exact algorithm, heuristic algorithm, and meta-heuristic algorithm [1,2]. Although the exact algorithm can obtain the optimal solution, its solving speed is slow and can can be used for small-sized problems; the heuristic algorithm is fast, simple, and efficient, but the solution results cannot reach the global optimal; and the meta-heuristic algorithm is relatively fast and effective, but the iterative search process is usually time-consuming and needs to be solved iteratively for each problem case. Moreover, these traditional optimization algorithms rarely make use of historical information to adjust behavior and cannot effectively use historical solving experience for learning; hence, there is great room for improvement in terms of solving large-scale problems.

In addition, different cases of problems have the same combinatorial optimization structure (the objective function or the coefficient of constraint conditions); there are only differences in specific values; for example, improving assembly workers' skills will reduce the working time of tasks, but the correlation between tasks does not change. As an artificial intelligence algorithm closer to the way of human thinking, the deep reinforcement learning algorithm has deep association learning ability compared with the traditional optimization algorithm. When solving the cases at a same scale, historical experience of different cases—that is, existing task assignment schemes—can be associated and learned, the essential information of problems can be mined, and task assignment strategies can be obtained and automatically updated to achieve efficient solutions to similar cases. Moreover, the deep reinforcement learning algorithm has higher adaptability to the complex and ever-changing production environment, which is easy to adjust so as to alter the solution. In addition, although deep reinforcement learning algorithms have begun to be tried in solving such AL balancing problems [6,7], they are currently used to optimize simulation models and resource allocation. Therefore, this paper carries out the study on load balancing of TAL based on deep reinforcement learning for the first time.

The remaining sections are introduced as follows. Section 2 provides a literature review for the load balancing-oriented TALBP and deep reinforcement learning. Section 3 shows the mathematical model of load balancing-oriented TALBP. Section 4 describes the deep reinforcement learning algorithm combining distributed proximal policy optimization (DPPO) and convolutional neural network (CNN). The experimental verification is carried out in Section 5 and conclusions and future work avenues are presented in Section 6.

2. Literature Review

2.1. Two-Sided Assembly Line Balance Oriented to Load Balancing

According to different production stages and research objectives, TALBPs are mainly divided into two categories [8]. The first type of balancing problem occurs in the design and planning stage, the aim was to explore that how to use the minimum number of workstations and/or stations to achieve production under the premise of a given cycle time. For this type problem and its variants, many scholars have designed exact algorithms [9,10], heuristic algorithms [11,12], and meta-heuristic algorithms [13,14]. With the assembly line officially put to use, some potential problems (task assignment, resource allocation, etc.) that could not be predicted in time in the design stage of the assembly line become increasingly obvious. Moreover, assembly workers' skills, product configuration, and customer demands may also change, and thus the task content and/or operation time changes and the original balancing plan needs to be further optimized. The second type of balance problem occurs in this stage, which is the problem of how to obtain a better cycle time by optimizing task assignment under the premise of definite workstations [15–17]. However, the above studies did not consider load balancing, which is the initial goal of balance research [5] and the source of maintaining the stable and efficient operation of the assembly line [18,19].

Considering the importance of load balancing, some scholars have specifically studied this kind of problem and defined it as the third type of balance problem—that is, how to optimize task assignment and balance the load of all stations as much as possible under the premise of fixed cycle time and line length. Ozcan and Toklu [20] used the assembly line smoothness index to measure load balancing and combined it with assembly line efficiency as optimization objectives to build a mathematical model of TAL load balancing. The minimum deviation method (MDM) was used to combine multi-objective problems into single-objective problems, and a tabu search algorithm was proposed. Lan and Yee [21] designed a nonlinear integer programming model to maximize the smoothness of the line for the third-type TALBP and solved it using Lingo. Purnomo and Wee [22] designed a harmonic search (HS), combining non-inferior sorting rules for TALBP considering zone constraints. The goal was to optimize the production efficiency and balance the load of stations. Li et al. [23] used an improved version of teaching and learning optimization (ITLBO) to balance multi-constraint and multi-objective TAL. The goals were the optimization of assembly line efficiency, assembly line smoothness, and total associated unit production costs. Li et al. [24] proposed an iterative local search algorithm (ILS) to realize the load balancing of TALs. A heuristic algorithm was applied to obtain the better initial population; the local search and disturbance factors were used iteratively until a local optimal solution was found. This algorithm also applies the priority decoding method based on the combination of orientation selection rules and task selection rules to reduce the load difference between the left and right stations of a workstation as far as possible so as to ensure that the sequence-related waiting time is reduced to a certain extent. Wu et al. [25] used a hybrid algorithm based on variable neighborhood search and gravity search for the TALBP considering zone constraints to achieve load balancing among workstations and reduce the series-dependent completion time of tasks as much as possible. Buyukozkan et al. [26] provided the dictionary bottleneck hybrid TALBP, balancing the load of all workstations by gradually minimizing the weighted load of the workstations with the maximum load and finally achieving the load balancing of all workstations on the assembly line. Yadav and Agrawal [27] measured load balancing by the idle time length on the workstations and established the related mathematical model. A branch and bound algorithm was programmed in the Lingo solver for this problem, and load balancing schemes of various benchmark problems were explored. After that, the workload maximization of stations was used as the measure of load balance [28], an exact algorithm was designed, and its effectiveness was verified through an engineering project. Abdullah Make and Ab Rashid [29] carried out a study on load balancing of TALs in automobile assembly workshops and designed a particle swarm optimization algorithm. To obtain a better task assignment scheme, in addition to cycle time and stations, the influences and constraints of workers' skill levels, tools, and equipment on assembly task assignment were also considered.

To sum up, current research on the load balancing of TALs is still mainly focused on the optimization design and solution of the traditional algorithm (i.e., exact, heuristic, and meta-heuristic algorithms), and as far as we know, there is no research on load balancing-oriented TALBP based on deep reinforcement learning, which is more suitable for complex and variable manufacturing processes. In addition, for multi-objective optimization, most of them are converted into single objectives by mathematical programming methods. The operation involved weighted coefficient or unified dimensions; that is, the search direction is delimited artificially, which reduces the search space and cannot guarantee the optimality of the solution.

2.2. Deep Reinforcement Learning

The reinforcement learning algorithm (RLA) could be used to obtain optimal strategies for sequencing decision problems [30]. As shown in Figure 2, the agent of RLA interacts with the environment continuously, observes the environment state s_t, makes a decision action a_t, and obtains the feedback (reward) r_t from the environment, then adjusts the strategy according to the feedback information from the environment so that the subsequent output decision action can meet the expectation.

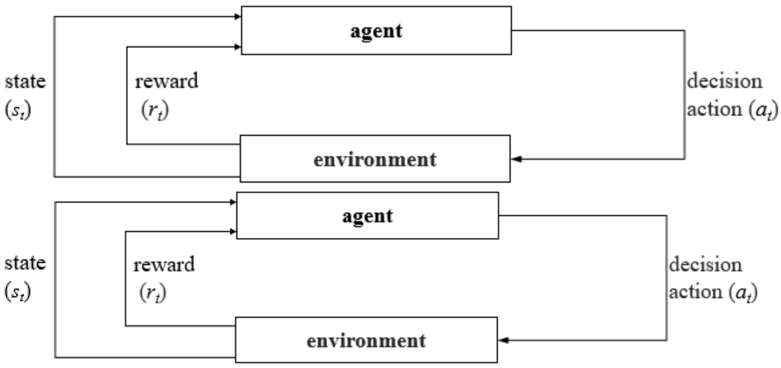


Figure 2. Solving process of reinforcement learning.

The deep learning algorithm (DLA) is a method by which to gradually acquire the whole picture of things through multi-level learning and abstracting from simple to complex [31]. The deep reinforcement learning algorithm (DRLA) generated by the combination of reinforcement learning (RL) and deep learning (DL) shows strong data processing and environment interaction abilities in terms of self-adaptation and self-learning, has rapid decision-making abilities combined with offline training and online decision-making, and highly versatile generalization ability, which can better solve complex problems [32].

In 2015, DeepMind [33] proposed the first DRLA—the Deep Q-learning Network (DQN). It combined deep neural networks with reinforcement learning and applied them to the research of games, Go (i.e., weiqi) and other fields, achieving impressive results. Since then, research on various deep reinforcement learning methods has been rapidly carried out, and the applications have expanded from games to other fields [34–36]. At present, for combinatorial optimization, deep reinforcement learning algorithms have already penetrated into the classic travel salesman problem [37], the path optimization problem [38], the packing problem [39], the maximum cutting problem [40], and other operational research problems. At the same time, some scholars, aiming to solve practical production problems, have also begun to introduce the deep reinforcement learning algorithm into the research of manufacturing problems [41]. In the area of AL balancing, Li et al. [6] focused on the research of balancing AL in the digital domain, and designed a DRLA with the support of deep deterministic policy gradient (DDPG) to enhance the operation and simulation effect of the assembly line digital twin model. Lv et al. [7] combined the sequencing problem with the assembly line balance problem and proposed a new version of DRLA on the basis of DDPG, in which an iterative interaction mechanism between task assembly time and station load were designed to achieve production task sequencing and worker allocation layer by layer. The objective was minimizing the work overload.

Although there have been some preliminary achievements in solving combinatorial optimization problems by DRLA, to the best of our knowledge, there are no studies of the deep reinforcement learning algorithm for the TALBP so far.

3. Mathematical Model for Load Balancing-Oriented TALBP

3.1. Assumptions

(1) The assembly line task assignment scheme is available, and the specific task assignment is known;

(2) The product variety is unique, and the processes are determined;

(3) Cycle time is deterministic;

(4) Execution time of takes and precedence relationship between tasks are known;

(5) Buffers, parallel stations and tasks are not considered.

3.2. Parameters

ns: Number of workstations.

nm: Number of stations.

I: Task set, $I = \{1,\ 2,\ \ldots,\ i,\ \ldots,\ m\}$.

J: Workstation set, $J = \{1,\ 2,\ \ldots,\ j,\ \ldots,\ n\}$.

(j, k): the specific station of the workstation j, i.e., the operating orientation of the station, k = 1, represents left station, k = 2, represents right station.

A_L: Task set that can only be assembled in the left station, $A_L \subset I$.

A_R: Task set that can only be assembled in the right station, $A_R \subset I$.

A_E: Task set that can be assembled in both left and right stations, $A_E \subset I$.

$P(i)$: Task set that contains all immediate precedence tasks of task i.

$P_a(i)$: Task set that contains all precedence tasks of task i.

$S(i)$: Task set that contains all immediate successor tasks of task i.

$S_a(i)$: Task set that contains all successor tasks of task i.

P_C: Set of tasks without precedence tasks.

$C(i)$: Task set opposite to operating orientation of task i. $C(i) = A_L, i \in A_R; C(i) = A_R$, $i \in A_L; C(i) = \Phi, i \in A_E$.

$K(i)$: Set of operating orientation indication of a task i. $K(i) = \{1\}, i \in A_L; K(i) = \{2\}$, $i \in A_R; K(i) = \{1,2\}, i \in A_E$.

t_i^s: Start time of the task i.

t_i^f: Finish time of the task i.

t_i: Operation time of the task i, $t_i = t_i^f - t_i^s$.

ct: Cycle time.

μ: A constant with a larger value.

$x_{ijk} = \{0,1\}$: If the task i is assigned to the workstation (j, k), the value is 1, otherwise it is 0.

$z_{ip} = \{0,1\}$: If task i and task p are assigned to the same workstation, the task i is assigned earlier than task p, the value is 1, otherwise 0.

3.3. Mathematical Model

Equations (1)–(3) are the objective functions, and their calculation formulae are shown in Equations (4)–(6). $ST_{jk} = \sum_{i \in S_{jk}} t_i$, is the total working time of tasks which are assigned to station (*j*,*k*), and $ST_{max} = max\{ST_{jk}\}$ is the maximum one of them. $Ct(j, k)$ represents the completion time of station (*j*,*k*), and $Ct_{max} = max\{Ct(j, k)\}$ is the maximum thereof. Equation (7) indicates that any task can only be assigned to one station. Equations (8) and (9) show cycle time constraints—that is, the completion time of each station must be less than cycle time. Equation (10) represents the precedence constraint. Equations (11)–(13) represent sequence-dependent constraints. Equations (14)–(17) give the definition of each variable.

$$\max LE \tag{1}$$

$$\min SI \tag{2}$$

$$\min CSI \tag{3}$$

$$LE = \frac{\sum_{i=1}^{nm} t_i}{ct * nm} \times 100\% \tag{4}$$

$$SI = \sqrt{\frac{\sum_{j \in J} \sum_{k=1}^{2} (ST_{\max} - ST_{jk})^2}{nm}} \tag{5}$$

$$CSI = \sqrt{\frac{\sum_{j \in J} \sum_{k=1}^{2} (Ct_{\max} - Ct(j,k)^2)}{nm}} \tag{6}$$

$$\sum_{j \in J} \sum_{k \in K(i)} x_{ijk} = 1, \forall i \in I \tag{7}$$

$$t_i^f \leq ct, \forall i \in I \tag{8}$$

$$t^f \geq t_i, \forall i \in I \tag{9}$$

$$\sum_{g \in J} \sum_{k \in K(i)} g x_{hjk} \leq \sum_{j \in J} \sum_{k \in K(i)} j x_{ijk}, \forall i \in I - P_0, h \in P(i) \tag{10}$$

$$t^f - t_h^f + \mu(1 - \sum_{k \in K(h)} x_{hjk}) + \mu(1 - \sum_{k \in K(i)} x_{ijk}) \geq t_i, \forall i \in I - P_0, h \in P(i), j \in J \tag{11}$$

$$\begin{gathered} t_p^f - t_i^f + \mu(1 - x_{pjk}) + \mu(1 - x_{ijk}) + \mu(1 - z_{ip}) \geq t_p, \forall i \in I, \\ p \in \{r | r \in I - (P_a(i) \cup S_a(i) \cup C(i)), i \prec r\}, j \in J, k \in K(i) \cup K(p) \end{gathered} \tag{12}$$

$$\begin{gathered} t_i^f - t_p^f + \mu(1 - x_{pjk}) + \mu(1 - x_{ijk}) + \mu z_{ip} \geq t_i, \forall i \in I, \\ p \in \{r | r \in I - (P_a(i) \cup S_a(i) \cup C(i)), i \prec r\}, j \in J, k \in K(i) \cup K(p) \end{gathered} \tag{13}$$

$$x_{ij1} = \{0, 1\}, i \in A_L, j \in J \tag{14}$$

$$x_{ij2} = \{0, 1\}, i \in A_R, j \in J \tag{15}$$

$$x_{ijk} = \{0, 1\}, i \in A_E, j \in J \tag{16}$$

$$z_{ip} = \{0, 1\}, \forall i \in I, p \in \{r | r \in I - (P_a(i) \cup S_a(i) \cup C(i)), i \prec r\} \tag{17}$$

4. Deep Reinforcement Learning Algorithm Based on DPPO and CNN (DPPO–CNN)

4.1. DPPO–CNN Agent

The architecture of DPPO–CNN with distributed multiple processes is shown Figure 3; the main process and m subprocess are turned on simultaneously. The main process is responsible for network training and update (gradient calculation and update), while the subprocess is only responsible for data acquisition without gradient calculation. The main process includes the experience pool and the main Actor–Critic network, and each subprocess includes the child Actor–Critic network. The Actor network is used to make task assignment decisions a_t based on the environmental status s_t of the TAL, while the Critic network is used to evaluate the quality of task allocation decisions a_t. Actor–Critic network structures are shown in Figures 4 and 5.

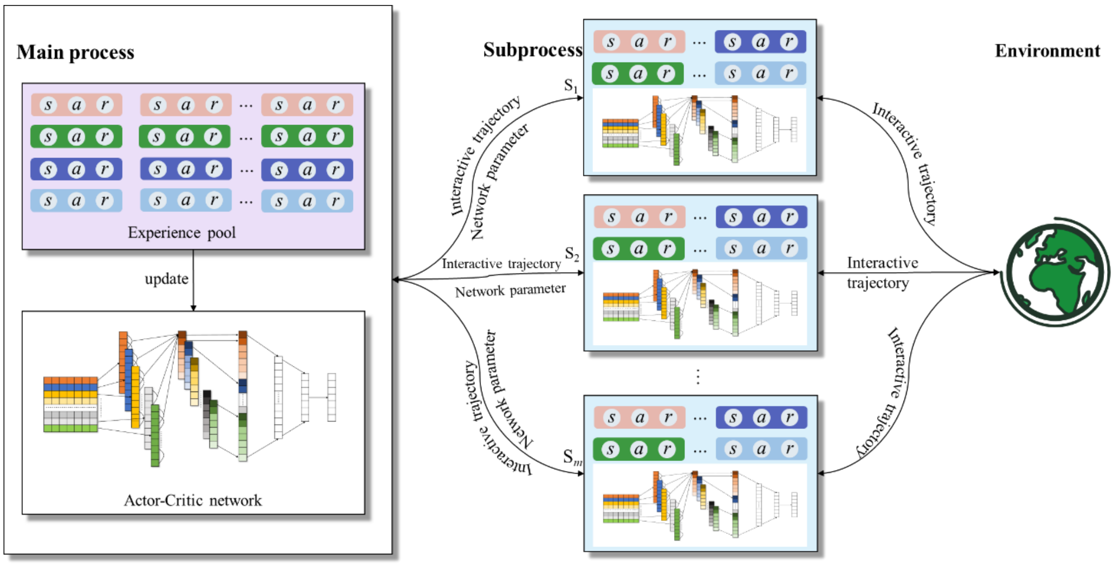


Figure 3. The architecture of DPPO–CNN.

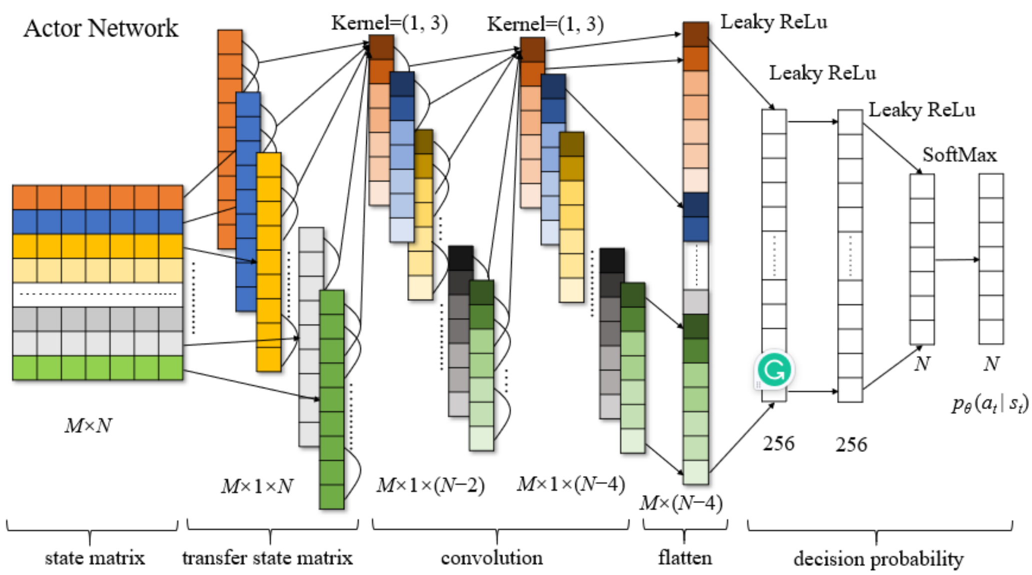


Figure 4. Actor network.

(1) Initialize the main Actor–Critic network of the main process and send its parameters to each subprocess through an orbit.

(2) The child Actor–Critic network of the subprocess loads the main Actor–Critic network and then interacts with the environment and transmits interactive trajectory (state matrix s, task assignment decisions a, reward function r) to the main process through an orbit.

(3) The main process stores the interaction experience of all subprocesses in the experience pool. When the amount of experience stored in the experience pool exceeds the capacity of the experience pool, it is packaged as a training set to train the main Actor–Critic network.

(4) Transfer the updated main Actor–Critic network to each subprocess again and go back to (1).

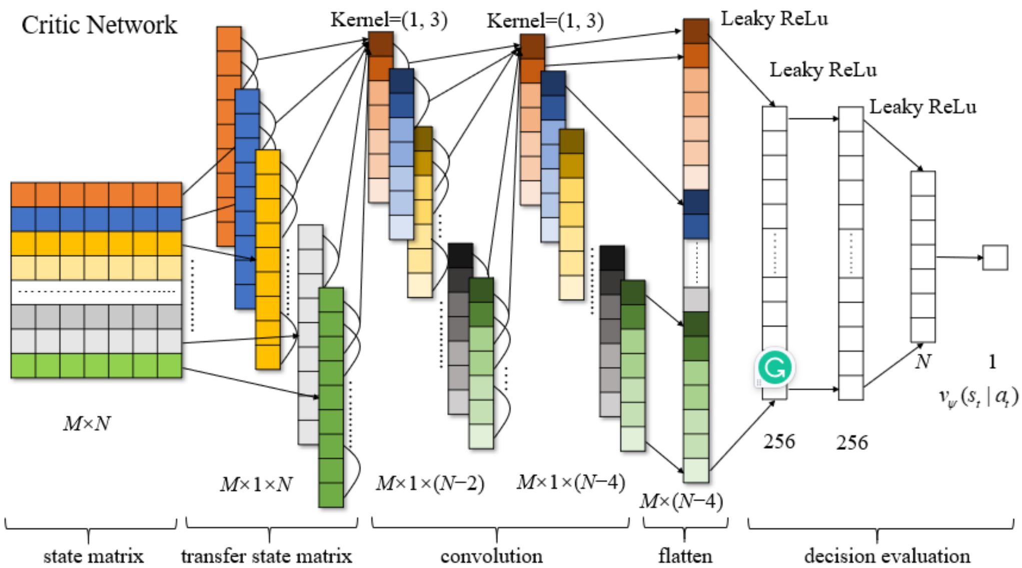


Figure 5. Critic network.

The Actor network is the strategy network, which approximates the optimal task assignment strategy $p_\theta(a_t|s_t)$ by using the neural network with parameter θ. The network structure is shown in Figure 4, including a two-layer convolutional network and a three-layer fully connected network. The dimensional matrix $M \times N$ is the input of the network at time t corresponding to the environmental status s_t. M is the number of feature vectors, and N is the number of total assembly tasks. After the matrix is input, two layers of convolution operations are carried out on it. The convolution Kernel size is as follows: Kernel = (1, 3). The feature vector obtained after convolution is flattened and input into the three fully connected layers. The number of nodes in the first two layers is 256, and the number of nodes in the last layer is N. Then, it is normalized by the SoftMax function and outputs $p_\theta(a_t|s_t)$, which is the probability of output task to be assigned a_t when the Actor policy network is in the status s_t.

The Critic network is evaluation network, which approximates the optimal strategy evaluation value $v_\Psi(s_t|a_t)$ by using the neural network with parameter Ψ. The network structure is shown in Figure 5. In this paper, the first several layers of the Critic network structure are the same as that of the Actor network structure, but the last layer is the linear regression layer; that is, the SoftMax layer in the Actor network is replaced with $v_\Psi(s_t|a_t) = f(h(t);\Psi) = \omega * h(t) + b$, where $v_\Psi(s_t|a_t)$ is the output of the Critic network at time t and $h(t)$ is the output of the previous fully connected layer. Ψ is the parameter of the internal unit node of the network, including the weight ω and the bias item b.

In each subprocess, the learning process is shown in Figure 6. The agent observes environmental status s_t of task assignment, makes the selection decision, and outputs the task to be assigned a_t and updates the status and gives back the reward r_t after the assignment of task a_t. The agent can solve the problem through continuous interaction with the environment. After each problem is solved, the trajectory of the interaction between the agent and the environment (including status, decision task, and reward) is stored in the experience pool. These dates are preprocessed to obtain training data in the experience pool. The interaction between the agent and the environment is suspended when the amount of data stored in the experience pool reaches its capacity limit. Parts of the training data are selected randomly by the agent from the experience pool, the network (task allocation strategy) is updated, and the experience is learned. The higher reward value is obtained through repeated trial and error, and eventually, the maximization of the cumulative reward and the optimization of the task allocation strategy can both be realized.

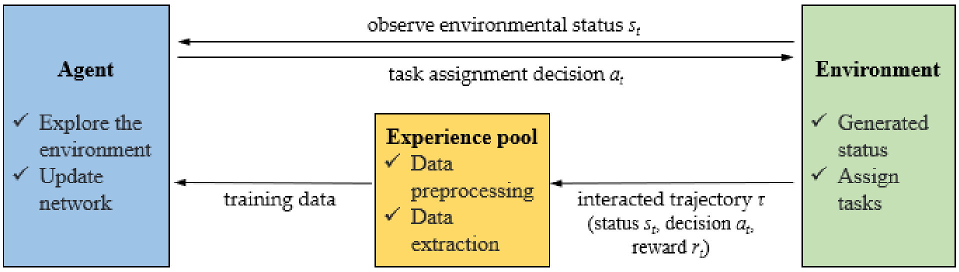


Figure 6. Learning process of each subprocess.

4.2. Task Assignment State of TAL

As shown in Table 1, there are 18 task assignment state features for the load balancing-oriented TALBP considered. Values of features 1–5 represent the related sequence numbers tasks; features 6–18 represent the overall situation of task assignment scheme of TAL; in particular, values of state features 11–16 are rounded.

Table 1. Task assignment state of TAL.

No.	Features	Description
1	PTime	A task with the longest operation time in the set of tasks without precedence tasks
2	MFlow	A task with the largest spare time caused by matched task in the set of tasks without precedence tasks
3	SucNum	A task with the largest number of successor tasks in the set of tasks without precedence tasks
4	AllSucNum	A task with the largest number of successor tasks
5	MTNum	A task with the smallest number of matched tasks in the set of tasks without precedence tasks
6	Side	The operating orientation of a task with the smallest sequence number in the set of tasks without precedence tasks (0 is E type of tasks; 1 is non-E type of tasks)
7	FAN	The number of tasks can be selected at present
8	NPN	The number of tasks without predecessors at present
9	NER	The number of remaining E type of tasks
10	NR	The number of remaining non-E type of tasks
11	LRDiffoverAE	The load difference of remaining non-E type of tasks/the average time of remaining E type of tasks
12	RToverAveptO	Remaining spare time of current station/the average time of tasks of current station
13	RToverAveptT	Remaining spare time of matched station/the average time of tasks of matched station
14	OPToverRemain	The operation time/remaining time of tasks
15	ARPW	Improved position weight
16	RRPW	Reverse position weight
17	PreNum	A task with the largest number of immediate successor tasks in the set of tasks without precedence tasks
18	AllPreNum	A task with the smallest number of precedence tasks

Original state feature information is abstracted and preprocessed by one-hot encoding to give it two-dimensional arrangement feature information, which is suitable for subsequent processing by convolutional neural network. If the number of tasks in TAL is N, the matrix with dimension $18 \times N$, as the environment state s_t, can be obtained after preprocessing.

Figure 7 shows the two types of the initial state matrix of P16 (Figure 8), and the state feature value of the initial state is shown in Table 2 (Figure 9).

(a)

Feature No. \ Task No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
15	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

Feature No. \ Task No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7. State matrix of P16: (**a**) state matrix s_1 generated for third column of Table 2; (**b**) state matrix s_2 generated for fourth column of Table 2.

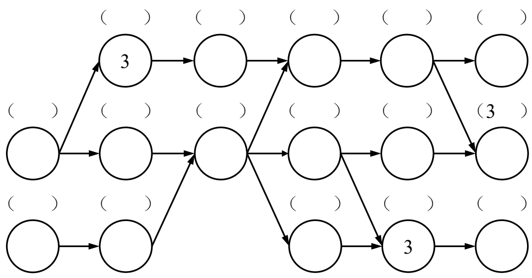


Figure 8. P16.

Table 2. Each state feature value of the initial state for P16.

No.	Features	State Feature Value (Task Assignment Scheme I)	State Feature Value (Task Assignment Scheme II)
1	PTime	1	1
2	MFlow	1	1
3	SucNum	1	1
4	AllSucNum	1	1
5	MTNum	1	1
6	ARPW	1	1
7	RRPW	1	1
8	PreNum	1	1
9	AllPreNum	1	1
10	Side	0	0
11	FAN	2	1
12	NPN	2	2
13	NER	10	9
14	NR	6	6
15	LRDiffoverAE	1	1
16	RToverAveptO	3	3
17	RToverAveptT	3	2
18	OPToverRemain	1	1

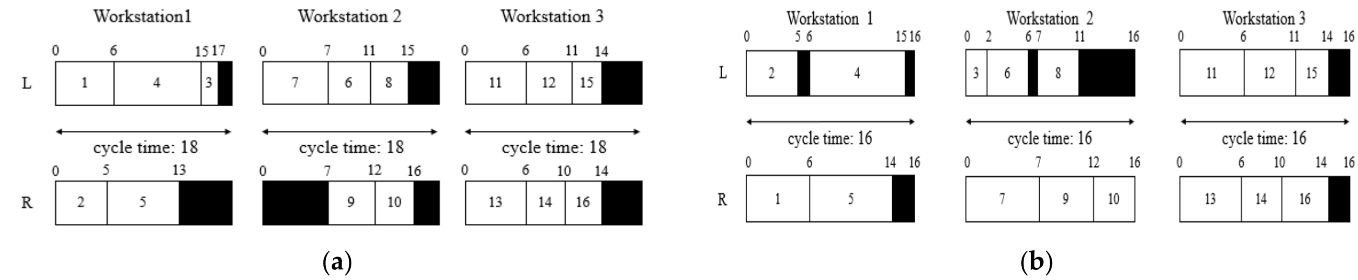


Figure 9. Task assignment schemes of P16: (**a**) task assignment scheme I; (**b**) task assignment scheme II.

4.3. Decision of Selecting Tasks

The mask layer is introduced to ensure that only the tasks that meet the constraints of precedence, operation orientation, and sequence dependence can be selected. Take the P16 problem as an example; the network parameters θ of the agent Actor strategy adopt orthogonal initialization. In the state s_1 of task assignment in the TAL environment, the output $p_\theta(a_1|s_1)$ is shown in Figure 9. At this time, if sampling is conducted according to the probability distribution, the task to be assigned is 3, which does not satisfy the precedence constraint. After processing at the mark layer—as shown in Figure 10—only task 1 and task 2 satisfy constraints, i.e., they can be selected. At this time, if sampling is conducted according to probability distribution, task 2 can be selected to be assigned.

probability	0.0604	0.0677	0.0727	0.0598	0.0653	0.0647	0.0634	0.0622	0.0650	0.0546	0.0594	0.0606	0.0607	0.0580	0.0639	0.0616
task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

mask

probability	0.0604	0.0677	~~0.0727~~	~~0.0598~~	~~0.0653~~	~~0.0647~~	~~0.0634~~	~~0.0622~~	~~0.0650~~	~~0.0546~~	~~0.0594~~	~~0.0606~~	~~0.0607~~	~~0.0580~~	~~0.0639~~	~~0.0616~~
task	1	2	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~

Figure 10. Mask layer process.

4.4. Task Assignment

Task assignment procedure is shown in Figure 11 and the contents are as follows.

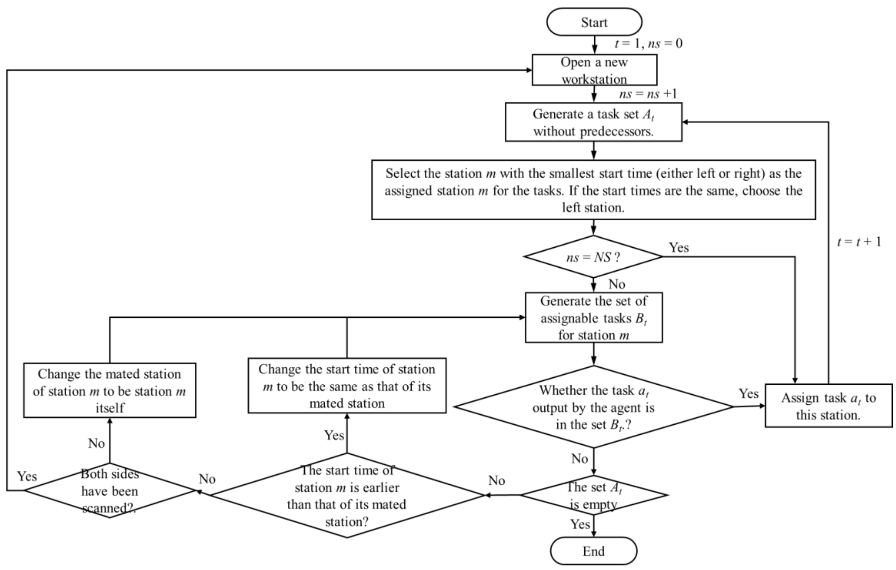


Figure 11. Task assignment.

Step 1: Initialize to generate the initial state of the TAL environment.

Step 2: Start a new workstation and set the earliest assignable task time of to 0.

Step 3: Generate a task set A_t without precedence tasks; alternatively, all of its precedence tasks have already been assigned, according to precedence constraint.

Step 4: Select the station (left or right) with a smallest start time as the assigned station *m*. If the start time is the same, choose the left station.

Step 5: Check whether the current workstation is the last one. If yes, perform Step 6; otherwise, turn to Step 7.

Step 6: Generate the set of assignable tasks B_t for station *m*.

Generate rules are as follows: (1) select the task without real-time predecessors; (2) select the task for which the completion time of its immediate predecessors is less than the earliest task-assignable time of the station; (3) select the task whose operating orientation is the same as the operating edge of station *m*.

Step 7. Check whether the task output by the agent is in the set B_t. If it is, turn to Step 11; otherwise, go to Step 8.

Step 8. If A_t is an empty set, stop this algorithm; otherwise, perform Step 9.

Step 9. If the earliest task assignable time of station (t_1) is earlier than the earliest task assignable time of the mated station (t_2), $t_1 = t_2$, and return to Step 5; otherwise, perform Step 10.

Step 10. If both sides of the station have been scanned, start a new workstation and return to Step 2; otherwise, scan the other side and return to Step 5.

Step 11. Assign the selected task by DPPO–CNN agent.

Step 12. Update the number of immediate predecessors and their related completion time of all the unassigned tasks, the earliest task assignable time of station *m*, and the assembly line state; then, return to Step 3.

4.5. Reward Function

Reinforcement learning agents can achieve the optimal task assignment strategy $p_\theta(a_t|s_t)$ by maximizing cumulative rewards (r_{sum}) and then realize the optimization objectives. Sparse reward is adopted by the traditional deep reinforcement learning algorithm, i.e., rewards r_1, r_2, ..., r_{n-1} are all equal to 0 until all tasks have been assigned. The environment gives feedback reward r_n to the agent; then, the accumulate reward $r_{sum} = r_n$. This is an easy way to converge the algorithm because for many tasks, the agent can obtain positive samples with a certain probability by conducting random exploration in the environment, and the positive samples occupy a relatively significant proportion of the total samples at the early stage of learning.

However, with the increase in task complexity, the probability of obtaining positive samples through random exploration becomes small, and the sparse feedback signals cannot indicate the exploration direction for the agent. The algorithm will be difficult to converge or the convergence speed will be very slow. To overcome this problem, it is necessary to add other reward items or punishment items to make the reward function become dense, and to guide the agent to explore the environment more efficiently [39]. In this paper, for objective *LE*, if both the operation time of the tasks and the number of stations are determined, the smaller the *ct*, the higher the *LE*. For objective *SI*, if the workloads of tasks which are assigned to a station are substantially equal to cycle time of that station, the difference between the $ST_{\max}$ and the ST_i will be small, and so will the *SI*. Similarly, for objective *CSI*, if the completion time of a station is substantially equal, the value of *CSI* is small, which means the operation of the assembly line is stable and balanced. Therefore, the values of SI and *CSI* will decrease if the idle time of each station is reduced.

There are two types of idle time that can be generated on a two-sided assembly line due to its parallel structure and the sequence-dependent relationship of its tasks, as shown in Figure 12: (1) End idle time—If the completion time of the current station will exceed the cycle time when the new task is assigned to the current station, the new task has to be assigned to the next station. In this case, the completion time of the current station will be less than the cycle time, and the time difference between the completion time of the current station and cycle time is the end idle time. As shown in Figure 12, on workstation 1, after task 5 is completed, task 4 needs to be assigned, but if the operation time of task 4 is 9, whether it is assigned to the left or right stations of workstation 1, the completion time will exceed cycle time 18; therefore, task 4 has to be assigned to workstation 2, and there has to be idle time at the end of the left and right stations of workstation 1. (2) Sequence-dependent idle time—If sequence-related tasks are assigned to the left and right stations of the same workstation, due to the absolute sequence constraint between them, this may lead to a situation in which some tasks may have to sit idle and wait, and this kind of idle time inside stations is called sequence-dependent idle time. As shown in Figure 12, on workstation 2, task 7, assigned to right station, can not be started from time 0 because its immediate precedent, task 4, is assigned to left station of the same workstation. Task 7 has to wait before task 4 is completed; therefore, there are nine time unitsof sequence-dependent idle time before task 7, which is caused by task 4.

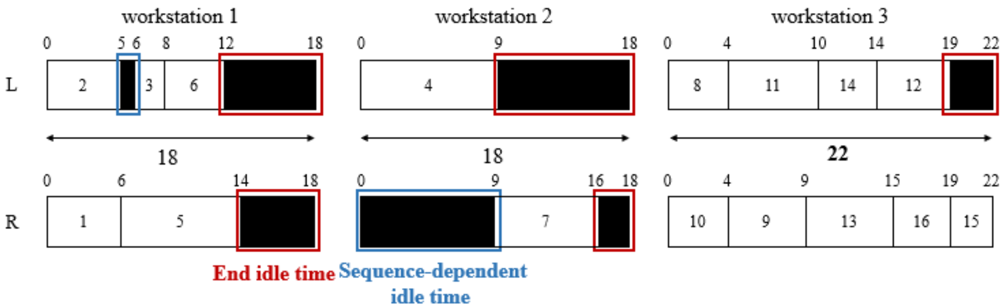


Figure 12. A task assignment scheme of P16.

Moreover, the logic of task assignment in this paper may cause the cycle time of the last workstation to be much larger than that of all other workstations, as shown in Figure 12. Due to fewer tasks being assigned to the first two stations, more tasks are assigned to the last workstation (3), resulting in the overloading of the completion time of workstation 3. Therefore, the difference between the actual cycle time of the last station (i.e., its completion time) and the ideal cycle time should be controlled.

Under these conditions, the reward function is set as follows during the task assignment process:

(1) If a task is the last task at the current station and there is end idle time at the current station, the reward function r is calculated as Equation (18):

$$r = -\ (end\ idle\ time\ of\ the\ current\ station); \tag{18}$$

(2) If there is idle time in the station caused by a between thesequence-dependent relationship of tasks, i.e., sequence-dependent idle time, the reward function r is calculated as Equation (19):

$$r = -\ (sequence - dependent\ idle\ time); \tag{19}$$

(3) If all tasks have been assigned in the last station, the reward function r is calculated as Equation (20):

$$r = -(\frac{le}{LE} + \frac{SI}{si} + \frac{CSI}{csi} + ct_{actual} - ct_{ideal}), \tag{20}$$

where ct_{actual} represents the actual cycle time of the last station (i.e., its completion time) and ct_{ideal} represents the ideal cycle time;

(4) The other state is equal to 0, as shown in Equation (21):

$$r = 0. \tag{21}$$

4.6. Overall Flow of DPPO–CNN

The overall flow of the proposed DPPO–CNN for TALBP-BL (Algorithm 1) is as follows.

Algorithm 1 Load balancing-oriented TALBP based on DPPO–CNN

1: Initializes the Actor–Critic network parameters θ, φ for the main process, maximum iteration number, experience pool capacity buffer size, maximum experience pool capacity max buffer size, batch data size, number of network updates of each round epoch; Subprocess Actor–Critic network parameters θ, φ.
2: **for** each *episode* **do**:
3: $t = 1$.
4: The main process empties the experience pool, sends its Actor–Critic network parameters θ, φ to each child process; each sub-process bilateral assembly line environment 1, 2, 3, . . . , m is initialized, generates states S_t^1, S_t^2, S_t^1, . . . , S_t^m; each agent 1, 2, 3, . . . , m loads the network parameters of the main process.
5: **while** *buffer size* < *max buffer size* **do:**
6: **while** all tasks of m's respective bilateral assembly lines have not been allocated **do**:
7: Agent 1, 2, 3, . . . , m observes the environment status S_t^1, S_t^2, S_t^1, . . . , S_t^m, respectively, and takes the tasks a_t^1, a_t^2, a_t^1, . . . , a_t^m to be assigned according to strategy $p_\theta(a_t|s_t)$.
8: Environment 1, 2, 3, . . . , m allocates tasks a_t^1, a_t^2, a_t^1, . . . , a_t^m separately, and feedback reward r_t^1, r_t^2, r_t^1, . . . , r_t^m.
9: $t = t + 1$.
10: Environment 1, 2, 3, . . . , m update states S_t^1, S_t^2, S_t^1, . . . , S_t^m.
11: **end while**
12: Agent 1, 2, 3, . . . , m stores the interactive trajectory (solving experience) τ_1, τ_2, τ_3, . . . , τ_m, respectively, in the experience pool and packages it as training data.
13: **end while**
14: **for** epoch in {1, 2, . . . , *epochs*} **do:**
15: Randomly extract the training data with the size of batch size from the experience pool.
16: Calculate the network loss function of the actor strategy of the master process; evaluate the critic loss function of the master process.
17: Update the main process Actor's policy network $p_\theta(a_t|s_t)$.
18: Update the main process Critic's evaluation network $v_\varphi(s_t, a_t)$.
19: **end for**
20: $\theta_{old}, \varphi_{old} \leftarrow \theta, \varphi$.
21: **end for**

5. Experimental Verification and Discussions

5.1. Implementation of the DPPO–CNN

In this paper, 59 instances of the benchmark problem (P9 [42] (Figure A1), P12 [42] (Figure A1), P16 [6] (Figure 8), P24 [42] (Figure A1), P65 [6] (Figure A2), P148 [5,6] (Table A1), and P205 [6] (Table A2)) are utilized to test the performance of the proposed DPPO–CNN. In addition, the DPPO–CNN algorithm is programmed in Python 3.6 and runs on a personal computer with Ubuntu 20.04LTS, 2.90 GHz CPU frequency, and the 16 g memory.

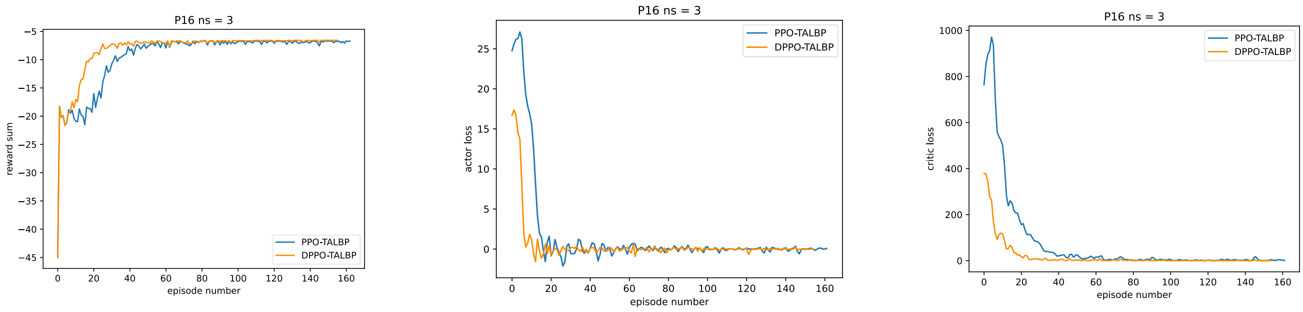
The parameters of DPPO–CNN are mainly decided according to the empirical values and the actual data of the agent interaction process, as listed in Table 3.

Table 3. Parameters of DPPO–CNN.

Parameter	Value	Parameter	Value
Number of hidden layers on the Actor and Critic network	256	Learning rate of the Actor network	10^{-4}
Activation function of hidden layers on the Actor and Critic network	Leaky Relu	Learning rate of the Critic network	2×10^{-4}
Activation function of output layers on the Actor network	Softmax	Sample size	256
Activation function of output layers on the Critic network	Leaky Relu	Maximum capacity of the experience pool	4096
Convolution kernel of convolution layers on the Actor and Critic network	(1, 3)	Study times per round	8
Initialize network parameters	Orthogonal initialization	Return discount factor	0.99
Optimizer	Adam	Cutting coefficient	0.2

5.2. Verification of DPPO–CNN in Term of Model Training

To verify the performance of distributed multiple processes of DPPO–CNN, the model training results of DPPO–CNN are compared with the deep reinforcement learning with single process, named the PPO–CNN, which uses the same parameters, state matrix, and reward function. To clarify the effect of distributed multiple processes, P16 is used as the representative of small-scale cases and P65 as the representative of large-scale cases for detailed explanation, as shown in Figures 13 and 14.



(**a**) (**b**) (**c**)

Figure 13. Comparison between DPPO–CNN and PPO–CNN in term of model training (P16): (**a**) cumulative reward curves; (**b**) loss curves of the Actor network; (**c**) loss curves of the Critic network.

Figure 13a shows the cumulative reward curves of DPPO–CNN and PPO–CNN for P16; we can see that: (1) the cumulative rewards of both DPPO–CNN and PPO–CNN are increasing gradually, and they converge after 60 rounds of training; this verifies the effectiveness of our algorithm indirectly; (2) the cumulative reward curve of DPPO–CNN algorithm is rising faster than that of PPO–CNN algorithm, which shows that DPPO–CNN is better than PPO–CNN; (3) the gap is not obvious because of P16 is relatively simple and both algorithms can obtain a good solution strategy quickly. However, from the related results of large-scale case P65 (Figure 14a,b), the advantages of DPPO–CNN are very prominent; the convergance of PPO–CNN needs 1200 rounds of training, whereas that of DPPO–CNN only needs 600 rounds. The study concludes that the distributed architecture designed in this paper is helpful in solving large-scale cases. A similar conclusion can be also obtained from the comparison between DPPO–CNN and PPO–CNN in terms of loss curves in P16 and P65, respectively, as shown in Figure 13b,c and Figure 14c,d.

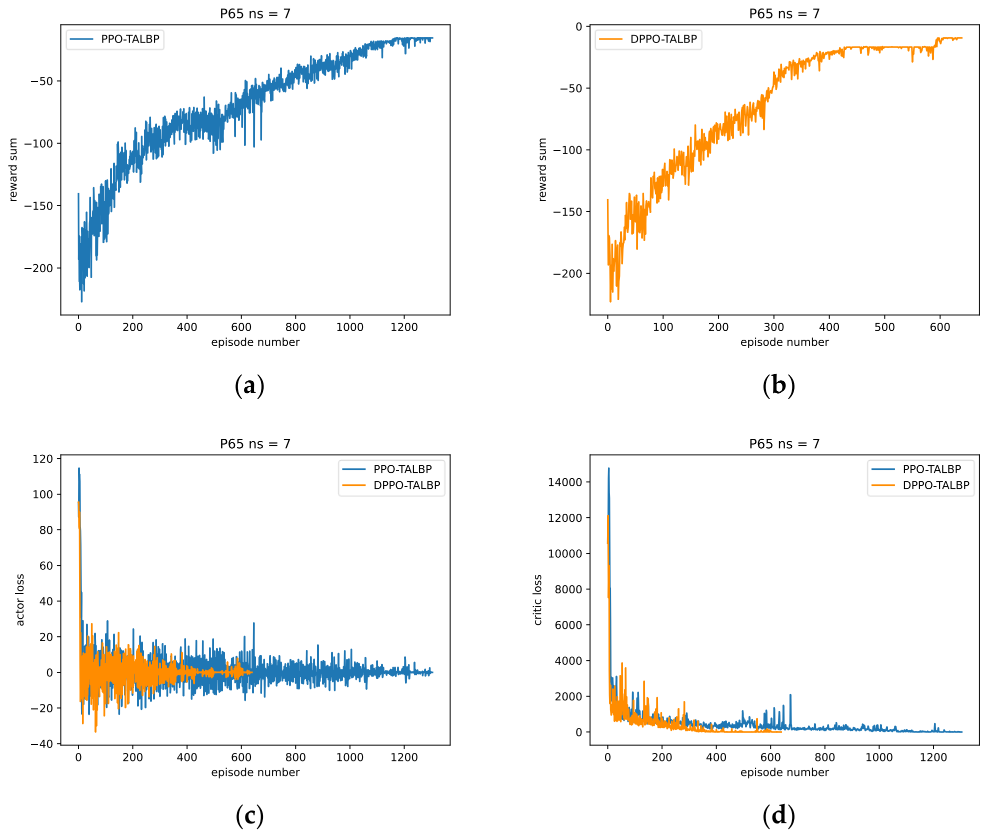


Figure 14. Comparison between DPPO–CNN and PPO–CNN in term of model training (P65): (**a**) cumulative reward curve of PPO–CNN; (**b**) cumulative reward curve of DPPO–CNN; (**c**) loss curves of the Actor network; (**d**) loss curves of the Critic network.

5.3. Verification of DPPO–CNN in Term of Solutions

The trained Actor–Critic network model of the DPPO–CNN agent is saved and utilized to solve the load balancing-oriented TALBP of all 59 instances with different scales. Both DPPO–CNN and PPO–CNN algorithms are ran 20 times for problem instance and the best results are record and exhibited in Table 4.

As can be seen from Table 4, the solution of the DPPO–CNN algorithm is superior to that of the PPO-TALBP algorithm in 49 cases out of all 59 test cases, which shows the absolute advantage of DPPO–CNN. Furthermore, (1) both DPPO–CNN and PPO–CNN perform well in solving the load balancing-oriented TALBP, especially in small-scale cases (P9, P12, P16, and P24), which shows that the main architecture of the deep reinforcement learning algorithm combining distributed proximal policy optimization (DPPO) and the convolutional neural network (CNN) is tenable; (2) DPPO–CNN has outstanding performance in solving large-scale cases (P65, P148, and P205). Take P148 with cycle time 255 as an example, *LE* = 91.34%, *SI* = 34.17, and *CSI* = 26.78 obtained by PPO–CNN, while *LE* = 99.53%, *SI* = 1.98, and *CSI* = 1.98 obtained by DPPO–CNN. Obviously, both SI and SCI decrease significantly, which means the load is more balanced. In addition, through calculation, it is found that in large-scale cases (P65, P148, and P205) that the solutions obtained by DPPO–CNN have an average increase of 7.89% in *LE*, an average decrease of 76.71% in *SI*, and an average decrease of 74.92% in *CSI* compared to the solutions obtained by PPO–CNN. (3) Although both DPPO–CNN and PPO–CNN perform excellently in terms of calculation time, the solution speed of DPPO–CNN is significantly better than that of PPO–CNN. For example, each kind of P65 instance can be resolved by DPPO–CNN within 0.04 s, while PPO–CNN resolves them 0.1 s; the calculation time here refers to the online solving time of the instance after the training is complete. Considering the fact that model training time makes up the majority of the running time of deep reinforcement learning algorithms, the comparison of the offline training time of the proposed DPPO–CNN and PPO–CNN is shown in Table 5. We can see that compared with PPO–CNN, DPPO–CNN also performs better in terms of offline training time.

Table 4. Result comparison.

Instance	*ct*	*ns*	PPO-CNN				DPPO–CNN			
			LE	*SI*	*CSI*	*Time(s)*	*LE*	*SI*	*CSI*	*Time(s)*
P9	3	3	94.44%	0.41	0.41	0.004	94.44%	0.41	0.41	0.001
	4	3	85.00%	1.87	1.87	0.004	94.44%	0.41	0.41	0.001
	5	2	85.00%	1.12	1.12	0.004	85.00%	1.12	1.12	0.001
	6	2	70.83%	2.50	1.58	0.005	85.00%	1.12	1.12	0001
	7	2	60.71%	3.57	3.57	0.005	85.00%	1.12	1.12	0.001
P12	4	4	78.13%	1.37	1.37	0.007	78.13%	1.37	1.37	0.002
	5	3	83.33%	1.23	0.91	0.007	83.33%	1.23	0.91	0.002
	6	3	83.33%	2.97	2.97	0.008	83.33%	1.23	0.91	0.002
	7	2	89.29%	1.12	1.12	0.008	89.29%	1.12	1.12	0.003
	8	2	78.13%	1.94	1.23	0.008	89.29%	1.12	1.12	0.003
	9	2	69.44%	3.35	1.87	0.009	89.29%	1.12	1.12	0.003
P16	15	4	78.095%	7.18	7.09	0.009	78.095%	7.18	7.09	0.003
	16	3	85.42%	2.94	2.52	0.009	85.42%	2.94	2.52	0.004
	18	3	75.93%	4.97	2.55	0.010	85.42%	2.94	2.52	0.004
	19	3	71.73%	5.86	2.38	0.011	85.42%	2.94	2.52	0.006
	20	3	68.33%	6.81	2.83	0.011	85.42%	2.94	2.52	0.006
	21	3	65.08%	8.56	6.18	0.012	85.42%	2.94	2.52	0.007
	22	2	93.18%	1.87	1.58	0.013	93.18%	1.87	1.58	0.007
P24	18	4	97.22%	0.71	0.61	0.014	97.22%	0.71	0.61	0.009
	20	4	87.5%	2.92	2.03	0.015	97.22%	0.71	0.61	0.009
	24	3	97.22%	1.00	0.71	0.017	97.22%	1.00	0.71	0.010
	25	3	95.33%	2.52	2.52	0.017	97.22%	1.00	0.71	0.010
	30	3	77.78%	7.92	5.21	0.020	97.22%	1.00	0.71	0.013
	35	2	100%	0.00	0.00	0.020	100%	0.00	0.00	0.014
	40	2	87.50%	6.82	2.55	0.020	100%	0.00	0.00	0.014
P65	326	8	97.76%	9.99	9.94	0.076	98.36%	8.79	8.79	0.024
	381	7	95.59%	27.42	25.03	0.079	98.98%	6.10	6.10	0.027
	435	6	97.68%	12.37	11.58	0.083	99.28%	4.59	4.59	0.030
	490	6	86.72%	102.11	40.07	0.087	99.28%	4.59	4.59	0.032
	512	5	99.59%	2.70	2.70	0.092	99.59%	2.70	2.70	0.037
	544	5	93.73%	43.76	36.76	0.100	99.59%	2.70	2.70	0.038
P148	204	13	96.61%	9.52	9.52	0.175	99.53%	1.66	1.66	0.094
	228	12	93.64%	22.61	17.38	0.181	99.30%	2.43	2.43	0.097
	255	11	91.34%	34.17	26.78	0.193	99.53%	1.98	1.98	0.101
	306	9	93.03%	32.58	20.76	0.211	99.35%	2.08	2.08	0.113
	357	8	89.71%	57.34	52.67	0.242	99.46%	2.18	2.18	0.118
	378	7	96.83%	20.44	5.09	0.256	99.73%	2.00	2.00	0.123
	408	7	89.71%	72.79	31.40	0.263	99.73%	2.00	2.00	0.127
	454	6	94.05%	40.96	13.82	0.271	99.77%	1.58	1.58	0.136
	459	6	93.03%	41.95	33.69	0.281	99.77%	1.58	1.58	0.144
	510	6	83.73%	133.08	128.98	0.297	99.77%	1.58	1.58	0.149
P205	1133	11	93.66%	103.255	76.98	0.761	98.44%	24.44	16.49	0.391
	1275	10	91.55%	157.90	108.27	0.783	98.34%	26.75	19.20	0.420
	1322	9	98.11%	35.00	29.71	0.789	98.70%	29.50	12.09	0.438
	1455	9	89.14%	202.97	119.14	0.821	98.70%	29.50	12.09	0.445
	1510	8	96.63%	63.64	34.84	0.843	98.85%	25.94	17.83	0.461
	1650	8	88.43%	265.00	96.17	0.855	98.85%	25.94	17.83	0.474
	1699	7	98.15%	46.30	19.52	0.857	98.79%	30.93	13.99	0.480
	1888	7	88.32%	374.85	253.90	0.864	98.79%	30.93	13.99	0.499
	1920	7	86.85%	447.20	371.80	0.869	98.79%	30.93	13.99	0.502
	2077	6	93.67%	197.45	121.67	0.888	98.90%	36.08	11.42	0.518
	2100	6	92.64%	205.35	157.83	0.891	98.90%	36.08	11.42	0.523
	2266	6	85.85%	459.64	395.98	0.907	98.90%	36.08	11.42	0.542
	2300	6	84.58%	516.95	456.03	0.912	98.90%	36.08	11.42	0.549
	2454	5	95.13%	219.16	63.68	0.920	99.08%	34.69	11.02	0.558
	2500	5	93.38%	251.86	112.88	0.944	99.08%	34.69	11.02	0.565
	2643	5	88.33%	419.17	277.99	0.952	99.08%	34.69	11.02	0.577
	2800	5	83.38%	683.82	659.15	0.981	99.08%	34.69	11.02	0.582
	2832	5	82.43%	735.43	711.63	0.992	99.08%	34.69	11.02	0.589

Table 5. Comparison of the offline training times of DPPO–CNN and PPO–CNN.

Instance	Training Time (h)	
	DPPO–CNN	PPO-CNN
P9	0.15	0.10
P12	0.20	0.12
P16	0.35	0.20
P24	0.50	0.35
P65	1.00	0.68
P148	1.50	0.95
P205	4.00	2.34

6. Conclusions and Future Work Avenues

In this article, the load balancing-oriented TALBP has been studied and a deep reinforcement learning algorithm combining distributed proximal policy optimization (DPPO) and convolutional neural network (CNN) has been presented. To the best of our knowledge, this is the first attempt to solve the load balancing-oriented TALBP based on deep reinforcement learning. A mathematical model with objectives of optimizing line efficiency (*LE*), smoothness index (*SI*), and completion time smoothness index (*CSI*) is provided. In the proposed deep reinforcement learning algorithm, distributed multiple processes have been proposed to improve the search speed and capability of solutions. A total of 18 task assignment state features for the load balancing-oriented TALBP environment have been considered, which ensure that the agent can obtain more useful information from the environment and perform optimal selection as completely as possible, and different reward functions according to objectives and their implicit information have been proposed to guide good solution direction. Fianlly, the performance of the proposed algorithm has been verified on all scales of benchmark instances via a comparison with the single-process deep reinforcement learning algorithm in terms of model training and solution results.

Although the proposed algorithm performs better in this study, there is still the possibility of improvement; for example, we could directly obtainin the Pareto-optimal solution set using the deep reinforcement learning algorithm and/or extend the application of the proposed algorithm to other versions of the TALBP or other types of assembly lines, such as linear and parallel assembly lines, as well as similar production planning problems, such as the two-sided disassembly line balancing problem [43], the assembly sequence problem [44], and so on.

Author Contributions: Conceptualization, G.J. and S.S.; methodology, G.J. and Y.Z.; validation, C.W. and B.L.; formal analysis, G.J. and C.W.; investigation, C.W.; resources, S.S. and B.L.; data curation, Y.Z.; writing—original draft preparation, G.J., S.S. and B.L.; writing—review and editing, Y.Z., X.H., and C.W.; supervision, X.H.; project administration, Y.Z. and X.H.; funding acquisition, Y.Z. and X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by China National Heavy Duty Truck Group Co., Ltd., (Project No. 22H010100926), and New young teachers research start-up Fund of Shanghai Jiao Tong University (Project No. 22X010503668).

Institutional Review Board Statement: Not applicable.

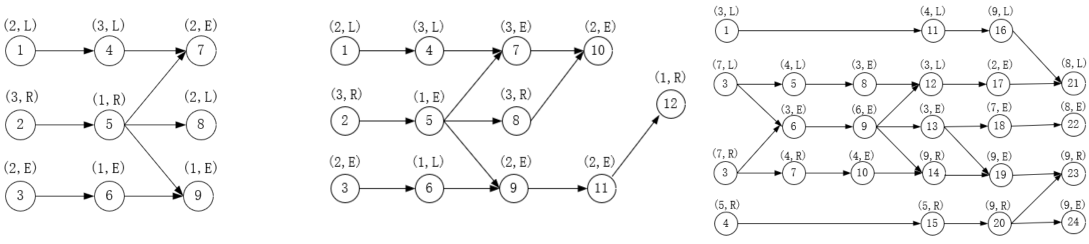
Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors are grateful to the anonymous reviewers, whose valuable comments and suggestions helped a lot to improve this paper, and to the editors for their kind and sincere support.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A



(**a**) (**b**) (**c**)

Figure A1. Small-scale cases: (**a**) P9; (**b**) P12; (**c**) P24.

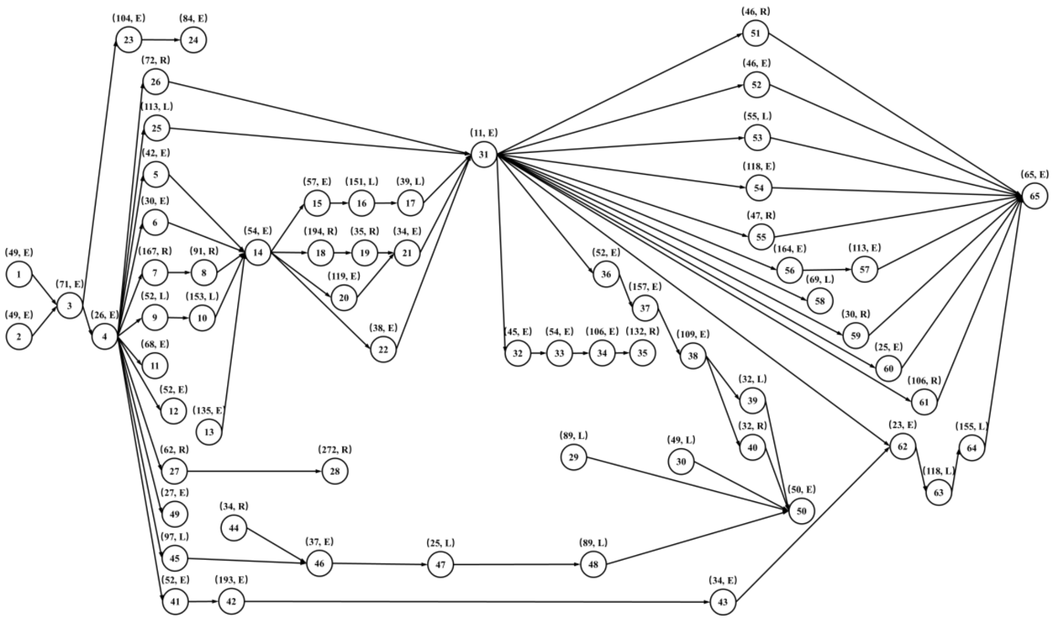


Figure A2. P65.

Table A1. P148.

NO.	Time	Side	Immediate Sucessors	NO.	Time	Side	Immediate Sucessors
1	16	E	5, 6, 7, 8	75	101	E	88, 97
2	30	E	3	76	5	E	77
3	7	E	4, 5, 6, 7	77	28	E	78
4	47	E	8	78	8	E	79
5	29	E	14	79	111	E	80
6	8	E	9	80	7	E	81
7	39	E	14	81	26	E	106
8	37	E	10	82	10	E	83, 89, 143, 146
9	32	E	14	83	21	E	
10	29	E	14	84	26	E	85
11	17	E	12	85	20	E	
12	11	E	13	86	21	E	
13	32	E		87	47	E	
14	15	E	15, 16	88	23	E	111
15	53	L	17	89	13	E	90
16	53	R	17	90	19	E	79
17	8	E	18, 19	91	115	E	105

Table A1. *Cont.*

NO.	Time	Side	Immediate Sucessors	NO.	Time	Side	Immediate Sucessors
18	24	L	20	92	35	E	135
19	24	R	20	93	26	L	
20	8	E	21, 22, 23, 24	94	46	E	
21	7	R	25, 26, 27, 28	95	20	E	101
22	8	L	25, 26, 27, 28	96	31	E	104
23	14	L	25, 26, 27, 28	97	19	E	
24	13	R	25, 26, 27, 28	98	34	E	101
25	10	R	29	99	51	E	100
26	25	R	29	100	39	E	101
27	11	L	29	101	30	E	102, 103
28	25	L	29	102	26	E	127
29	11	E	31	103	13	E	127
30	29	R		104	45	E	
31	25	E	36	105	58	E	119
32	10	L	34	106	28	E	107
33	14	R	35	107	8	E	108
34	41	L	36	108	43	E	109
35	42	R	36	109	40	E	110
36	47	R	37	110	34	E	
37	7	R	38, 45	111	23	E	112
38	80	R	39	112	162	L	113
39	7	R	40	113	11	L	114, 116, 120, 123, 128
40	41	R	41, 48, 55	114	19	E	115
41	47	R		115	14	E	125
42	16	L	43	116	31	E	117
43	32	L	44	117	32	E	118
44	66	L		118	26	E	126
45	80	L	46	119	55	E	
46	7	L	47	120	31	E	121
47	41	L	48, 49, 55	121	32	E	122
48	13	E		122	26	E	126
49	47	L		123	19	E	124
50	33	E	51	124	14	E	125
51	34	L	53,69	125	19	E	
52	11	L	53	126	48	E	
53	118	L		127	55	E	
54	25	L	133	128	8	L	129
55	7	R	54, 72, 76, 87, 88	129	11	L	130
56	28	E	73	130	27	L	131, 137
57	12	L	79	131	18	L	
58	52	L	84, 86	132	36	E	135
59	14	E	75, 87	133	23	L	135
60	3	E		134	20	R	135
61	3	E	62	135	46	E	136
62	8	E	63	136	64	E	
63	16	E	67	137	22	L	
64	33	R	65, 71, 72	138	15	E	139
65	8	E	66,99	139	34	E	140
66	18	E	67	140	22	E	
67	10	E	68	141	151	L	142
68	14	E	95,98	142	148	R	143, 146, 147, 148
69	28	R	82	143	64	L	
70	11	R	71	144	170	L	145
71	118	R		145	137	R	147, 148
72	25	R	134	146	64	R	
73	40	E	84, 86, 87, 88, 96	147	78	L	
74	40	E	75	148	78	R	

Table A2. P205.

NO.	Time	Side	Immediate Sucessors	NO.	Time	Side	Immediate Sucessors
1	692	E	36	104	68	R	113
2	42	E	3, 4	105	232	L	106, 107
3	261	R	5	106	122	L	108
4	261	L	5	107	151	E	108
5	157	E	7, 13	108	31	L	113
6	90	E	36	109	97	E	113
7	54	R	8	110	308	R	113
8	67	R	9	111	116	L	113
9	30	R	10	112	312	R	113
10	106	R	11	113	34	E	114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 161, 162, 163, 169, 171, 174, 203, 204, 205
11	32	R	12	114	128	L	160
12	62	R	36	115	54	E	160
13	54	L	14	116	175	R	160
14	67	L	15	117	55	E	160
15	30	L	16	118	306	E	126
16	106	L	17	119	59	E	126
17	32	L	18	120	59	E	126
18	62	L	36	121	66	E	126
19	56	E	36	122	66	E	126
20	67	E	22	123	23	E	126
21	86	E	22	124	244	E	125
22	37	E	23	125	54	E	126
23	41	E	24, 34	126	294	R	127, 128, 129
24	72	E	26, 27, 28	127	84	E	135
25	86	R	28	128	61	E	135
26	16	L	35	129	57	E	135
27	51	R	35	130	38	R	136
28	66	R	29	131	944	E	132
29	41	R	30, 33	132	511	R	133
30	72	R	31, 32	133	625	R	189
31	51	R	35	134	445	R	189
32	16	R	35	135	68	L	136, 137, 138, 139, 140, 141, 142, 144, 145, 147, 148, 149, 150, 151, 152, 153, 158
33	15	R	35	136	53	L	189
34	15	L	35	137	49	E	160
35	85	E	36	138	92	E	160
36	59	E	37, 40, 41, 42, 62, 69, 72, 75, 83, 110, 111, 112	139	236	E	160
37	23	L	38	140	116	L	143
38	13	L	39	141	265	L	143
39	19	L	45	142	149	L	143
40	108	E	43, 54	143	74	L	160
41	214	E	92	144	332	E	160
42	80	E	43, 54	145	324	E	146
43	37	L	44	146	104	L	160
44	84	L	45	147	51	L	160
45	18	L	46, 48, 51, 53	148	58	R	160
46	12	L	47	149	67	R	160
47	29	L	92	150	49	R	160
48	37	L	49	151	107	E	160
49	13	L	50	152	38	L	160
50	70	L	92	153	27	L	154
51	217	L	52	154	68	E	155
52	72	L	92	155	207	E	156
53	85	L	92	156	202	E	157

Table A2. *Cont.*

NO.	Time	Side	Immediate Sucessors	NO.	Time	Side	Immediate Sucessors
54	43	R	55	157	83	E	189
55	97	R	56, 59, 61	158	35	R	159
56	37	R	57	159	58	R	189
57	13	R	58	160	42	E	164, 170, 178, 179, 184
58	35	R	92	161	68	R	167
59	217	R	60	162	68	R	165
60	72	R	92	163	68	R	164
61	85	R	92	164	103	R	165
62	25	E	63	165	103	R	166
63	37	E	64	166	103	R	167
64	37	E	65, 68	167	103	R	168
65	103	E	66	168	103	R	177
66	140	E	67	169	68	L	170
67	49	E	80	170	103	L	172
68	35	E	80	171	68	L	172
69	51	E	70	172	103	L	173
70	88	E	71	173	103	L	175
71	53	E	73	174	68	L	175
72	144	E	73	175	103	L	176
73	337	E	74	176	103	L	177
74	107	E	76	177	10	E	185, 186, 187, 188, 194, 195
75	371	E	92	178	187	E	180
76	97	E	77, 78, 79	179	134	L	180
77	166	E	80, 82	180	89	L	181, 183
78	92	L	80	181	58	L	182
79	92	R	80	182	49	L	
80	106	E	81	183	134	L	
81	49	E	84	184	53	L	
82	92	E	92	185	334	E	189
83	371	E	92	186	24	R	189
84	87	E	85	187	76	R	189
85	162	E	86, 88, 90	188	76	L	189
86	96	E	87	189	192	E	190, 191, 193
87	79	E	92	190	98	E	
88	96	E	89	191	258	R	192
89	42	E	92	192	165	E	
90	88	R	91	193	38	R	
91	90	R	92	194	115	E	197
92	97	R	93, 94, 95, 96, 97, 98, 99	195	83	L	196
93	270	R	135	196	56	R	197
94	452	E	135	197	29	R	198, 199, 201
95	48	R	113	198	303	R	
96	338	E	113	199	18	R	200
97	34	E	100	200	29	R	
98	65	E	100	201	154	L	202
99	50	E	100	202	90	L	
100	112	E	101, 103, 105, 109, 130, 131, 134	203	93	L	
101	48	E	102	204	94	E	
102	117	E	113	205	165	E	
103	50	E	104				

References

1. Zhang, Y.; Hu, X.; Wu, C. A modified multi-objective genetic algorithm for two-sided assembly line re-balancing problem of a shovel loader. *Int. J. Prod. Res.* **2018**, *56*, 3043–3063. [CrossRef]
2. Zhang, Y.; Hu, X.; Wu, C. Improved imperialist competitive algorithms for rebalancing multi objective two-sided assembly lines with space and resource constraints. *Int. J. Prod. Res.* **2020**, *58*, 3589–3617. [CrossRef]

3. Hu, X.; Wu, E.; Bao, J.; Jin, Y. A Branch-and-bound Algorithm to Minimize the Line Length of a Two-sided Assembly Line. *Eur. J. Oper. Res.* **2010**, *206*, 703–707.
4. Hu, X. *Two-Sided Assembly Line Balancing Algorithm and Its Application*, 1st ed.; Science Press: Beijing, China, 2015; pp. 3–10.
5. Bartholdi, J.J. Balancing Two-sided Assembly Lines: A Case Study. *Int. J. Prod. Res.* **1993**, *31*, 2447–2461. [CrossRef]
6. Li, J.; Pang, D.; Zheng, Y.; Le, X. Digital Twin Enhanced Assembly Based on Deep Reinforcement Learning. In Proceedings of the 11th International Conference on Information Science and Technology (ICIST), Chengdu, China, 21–23 May 2021; pp. 432–437.
7. Lv, Y.; Tan, Y.; Zhong, R.; Zhang, P.; Wang, J.; Zhang, J. Deep reinforcement learning-based balancing and sequencing approach for mixed model assembly lines. *IET Coll. Intell. Manuf.* **2022**, *4*, 181–193. [CrossRef]
8. Lee, T.O.; Kim, Y.; Kim, Y.K. Two-sided assembly line balancing to maximize work relatedness and slackness. *Comput. Ind. Eng.* **2001**, *40*, 273–292. [CrossRef]
9. Hu, X.; Wu, E.; Jin, Y. A station-oriented enumerative algorithm for two-sided assembly line balancing. *Eur. J. Oper. Res.* **2008**, *186*, 435–440. [CrossRef]
10. Wei, N.; Liu, S.; Chen, C.; Xu, Y.; Shih, Y. An Integrated Method for Solving the Two-Sided Assembly Line Balancing Problems. *J. Adv. Manuf. Syst.* **2023**, *22*, 181–203. [CrossRef]
11. Li, Z.; Tang, Q.; Zhang, L. Two-sided assembly line balancing problem of type I: Improvements, a simple algorithm and a comprehensive study. *Comput. Oper. Res.* **2016**, *79*, 78–93. [CrossRef]
12. Álvarez-Miranda, E.; Pereira, J.; Vargas, C.; Vilà, M. Variable-depth local search heuristic for assembly line balancing problems. *Int. J. Prod. Res.* **2023**, *61*, 3102–3120. [CrossRef]
13. Yılmaz, D.; Emel, K.A.; Uğur, Ö.; Mehmet, S.İ. A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. *J. Intell. Manuf.* **2017**, *28*, 23–36.
14. Huang, D.; Mao, Z.; Fang, K.; Yuan, B. Combinatorial Benders decomposition for mixed-model two-sided assembly line balancing problem. *Int. J. Prod. Res.* **2022**, *60*, 2598–2624. [CrossRef]
15. Kim, Y.K.; Song, W.S.; Kim, J.H. A Mathematical Model and a Genetic Algorithm for Two-sided Assembly Line Balancing. *Comput. Oper. Res.* **2009**, *36*, 853–865. [CrossRef]
16. Lei, D.; Guo, X. Variable neighborhood search for the second type of two-sided assembly line balancing problem. *Comput. Oper. Res.* **2016**, *72*, 183–188. [CrossRef]
17. Kang, H.; Lee, A.H.I. An evolutionary genetic algorithm for a multi-objective two-sided assembly line balancing problem: A case study of automotive manufacturing operations. *Qual. Technol. Quant. Manag.* **2023**, *20*, 66–88. [CrossRef]
18. Azizoglu, M.; Imat, S. Workload smoothing in simple assembly line balancing. *Comput. Oper. Res.* **2018**, *89*, 51–57. [CrossRef]
19. Walter, R.; Schulze, P. On the performance of task-oriented branch-and-bound algorithms for workload smoothing in simple assembly line balancing. *Int. J. Prod. Res.* **2022**, *60*, 4654–4667. [CrossRef]
20. Uğur, Ö.; Bilal, T. A tabu search algorithm for two-sided assembly line balancing. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 822–829.
21. Lan, C.H.; Yee, M.S. Construct an INLP Mathematical Model to solve the Two-sided Assembly Line Balancing problem of Type-3. *Adv. Mater. Res.* **2012**, *383–390*, 4302–4305. [CrossRef]
22. Purnomo, H.D.; Wee, H.M. Maximizing production rate and workload balancing in a two-sided assembly line using Harmony Search. *Comput. Ind. Eng.* **2014**, *76*, 222–230. [CrossRef]
23. Li, D.; Zhang, C.; Shao, X.; Lin, W. A multi-objective TLBO algorithm for balancing two-sided assembly line with multiple constraints. *J. Intell. Manuf.* **2016**, *7*, 725–739. [CrossRef]
24. Li, Z.; Tang, Q.; Zhang, L.; Hu, J. Balancing two-sided assembly line with a simple and effective iterated local search algorithm. *ICIC Express Lett.* **2015**, *9*, 2695–2701.
25. Wu, Y.; Tang, Q.; Zhang, L. A hybrid gravitational search algorithm for two-sided assembly line balancing problem with zoning constraints. *ICIC Express Lett. Part B Appl.* **2016**, *7*, 2633–2640.
26. Buyukozkan, K.; Kucukkoc, I.; Satoglu, S.I.; Zhang, D.Z. Lexicographic bottleneck mixed-model assembly line balancing problem-Artificial bee colony and tabu search approaches with optimised parameters. *Expert Syst. Appl.* **2016**, *50*, 151–166. [CrossRef]
27. Yadav, A.; Agrawal, S. Minimize idle time in two sided assembly line balancing using exact search approach. In Proceedings of the 2019 International Conference on Management Science and Industrial Engineering, Phuket, Thailand, 24–26 May 2019; pp. 220–227.
28. Yadav, A.; Verma, P.; Agrawal, S. Mixed model two sided assembly line balancing problem: An exact solution approach. *Int. J. Syst. Assur. Eng.* **2020**, *11*, 335–348. [CrossRef]
29. Abdullah Make, M.R.; Ab Rashid, M.F.F. Optimization of two-sided assembly line balancing with resource constraints using modified particle swarm optimisation. *Sci. Iran.* **2022**, *29*, 2084–2098.
30. Chen, X. Research on Scheduling and Navigation Strategies Based on Reinforcement Learning. Master's Thesis, Zhejiang University, Hangzhou, China, 2021.
31. Tercan, H.; Meisen, T. Machine learning and deep learning based predictive quality in manufacturing: A systematic review. *J. Intell. Manuf.* **2022**, *33*, 1879–1905. [CrossRef]
32. Li, C.; Zheng, P.; Yin, Y.; Wang, B.; Wang, L. Deep reinforcement learning in smart manufacturing: A review and prospects. *CIRP J. Manuf. Sci. Technol.* **2023**, *40*, 75–101. [CrossRef]
33. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]

34. Chen, X.; Yao, L.; McAuley, J.; Zhou, G.; Wang, X. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowl.-Based Syst.* **2023**, *264*, 110335. [CrossRef]
35. Goby, N.; Brandt, T.; Neumann, D. Deep reinforcement learning with combinatorial actions spaces: An application to prescriptive maintenance. *Comput. Ind. Eng.* **2023**, *179*, 109165. [CrossRef]
36. Kallestad, J.; Hasibi, R.; Hemmati, A.; Sorensen, K. A general deep reinforcement learning hyper heuristic framework for solving combinatorial optimization problems. *Eur. J. Oper. Res.* **2023**, *309*, 446–468. [CrossRef]
37. Zhang, Z.; Liu, H.; Zhou, M.; Wang, J. Solving Dynamic Traveling Salesman Problems With Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn.* **2023**, *34*, 2119–2132. [CrossRef] [PubMed]
38. De Sirisuriya, S.C.M.S.; Fernando, T.G.I.; Ariyaratne, M.K.A. Algorithms for path optimizations: A short survey. *Computing* **2023**, *105*, 293–319. [CrossRef]
39. Jiang, Y.; Cao, Z.; Zhang, J. Learning to Solve 3-D Bin Packing Problem via Deep Reinforcement Learning and Constraint Programming. *IEEE Trans. Cybern.* **2023**, *53*, 2864–2875. [CrossRef]
40. Dai, H.; Khalil, E.B.; Zhang, Y.; Dilkina, B.; Song, L. Learning combinatorial optimization algorithms over graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6349–6359.
41. Zhang, C.; Wu, Y.; Ma, Y.; Song, W.; Le, Z.; Cao, Z.; Zhang, J. A review on learning to solve combinatorial optimisation problems in manufacturing. *IET Collab. Intell. Manuf.* **2023**, *5*, e12072. [CrossRef]
42. Kim, Y.K.; Kim, Y.; Kim, Y.J. Two-sided assembly line balancing: A genetic algorithm approach. *Prod. Plan. Control* **2000**, *11*, 44–53. [CrossRef]
43. Wang, K.; Li, X.; Gao, L. A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. *Comput. Ind. Eng.* **2019**, *130*, 634–649. [CrossRef]
44. Raju Bahubalendruni, M.V.A.; Biswal, B.B. An efficient stable subassembly identification method towards assembly sequence generation. *Natl. Acad. Sci. Lett.* **2018**, *41*, 375–378. [CrossRef]

applied sciences

MDPI

Article

Production Planning Forecasting System Based on M5P Algorithms and Master Data in Manufacturing Processes

Hasup Song [1,2], Injong Gi [1,2], Jihyuk Ryu [1,2], Yonghwan Kwon [3] and Jongpil Jeong [1,*]

[1] Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro Jangan-gu, Suwon 16419, Republic of Korea; shs5107@g.skku.edu (H.S.)
[2] ThiraUtech SCM Labs, Hakdong-ro 5-gil Gangnam-gu, Seoul 06044, Republic of Korea
[3] Mechanical Engineering, Karlsruhe Institute of Technology, 76049 Karlsruhe, Baden-Württemberg, Germany
[*] Correspondence: jpjeong@skku.edu; Tel.: +82-31-299-4267

Abstract: With the increasing adoption of smart factories in manufacturing sites, a large amount of raw data is being generated from manufacturers' sensors and Internet of Things devices. In the manufacturing environment, the collection of reliable data has become an important issue. When utilizing the collected data or establishing production plans based on user-defined data, the actual performance may differ from the established plan. This is particularly so when there are modifications in the physical production line, such as manual processes, newly developed processes, or the addition of new equipment. Hence, the reliability of the current data cannot be ensured. The complex characteristics of manufacturers hinder the prediction of future data based on existing data. To minimize this reliability problem, the M5P algorithm, is used to predict dynamic data using baseline information that can be predicted. It combines linear regression and decision-tree-supervised machine learning algorithms. The algorithm recommends the means to reflect the predicted data in the production plan and provides results that can be compared with the existing baseline information. By comparing the existing production plan with the planning results based on the changed master data, it provides data results that help production management determine the impact of work time and quantity and confirm production plans. This means that forecasting data directly affects production capacity and resources, as well as production times and schedules, to help ensure efficient production planning.

Keywords: production planning; predictive modeling; master data; machine learning; M5P Algorithm

1. Introduction

Supply chain management (SCM) is being studied and rapidly applied to manufacturing floors, where artificial intelligence (AI) can provide visibility and transparency for rapid and responsive decision making. This research helps improve quality control, reduce defects, and increase customer satisfaction [1]. Many studies have been conducted to identify the contribution of AI to SCM through systematic reviews of manufacturing systems [2]. Production planning and scheduling are at the core of SCM. These are important problems in various industries and require efficient scheduling methods to improve productivity and reduce costs [3]. Researchers have been working on solving workshop scheduling problems using machine learning algorithms, and their contributions can be seen. These studies show that there is a need for research to improve the production planning process by applying new algorithms in various fields. This research aims to investigate how technology can be applied to production planning in the field of supply chain management.

Modern manufacturing systems are increasingly complex, dynamic, and connected. Recent advances in AI, particularly machine learning, have shown great potential to transform the manufacturing sector, as the myriad of uncertainties and interdependencies make factory operations highly non-linear and stochastic [4]. It is contended that that AI can be used to improve the quality of data, optimize processes, and make better decisions [5]. Since the Industrial Revolution, the mass production system has been discarded completely in favor of multi-product, low-volume production to satisfy the complex and diverse needs of customers. Moreover, the life-cycle of products has been shortened. The complexity of the manufacturing process from demand to production to shipment has increased exponentially with technological advances. Manufacturers have factories in multiple locations that provide products through multiple supply chains. In this increasingly complex manufacturing environment with multiple tradeoffs, production planning within a given capacity should be performed appropriately to satisfy various constraints and demands. In the current competitive and dynamic business environment, reliable data has become even more important with the advancement of Internet of Things (IoT) and cloud technology. Furthermore, predicting and utilizing data from numerous sources remains a key challenge in the current situation.

Production planning has a wide range of influences on SCM. The objective of SCM is to improve efficiency, quality, productivity, and customer satisfaction. It plays an important role in achieving these objectives [6]. Production planning optimizes the use of raw materials and resources, coordinates production volumes and schedules, and efficiently manages the production process. It can help identify and improve problems in areas such as inventory management, production line efficiency, and productivity. Thereby, it provides factors that can have a positive impact [7].

It learns baseline information data (which form the basis of production planning according to performance data) and makes predictions using machine learning, compares with and reflects on existing master data, establishes plans through the production planning system, and provides indicators to help manage production.

Section 1 presents the background and purpose of the study, as well as the methodology and organization of the study. Section 2 describes the importance of master data from the perspective of the manufacturing industry before outlining the contents of the study. Section 3 describes the algorithms applied from a technical point of view and discusses previous research. Section 4 describes the design of the process. Section 5 provides the experimental results and discussion. Section 6 summarizes the results, presents the limitations, and considers directions for future research.

2. Related Work

2.1. The Importance of Managing Manufacturer Master Data

With the development of technology and the application of the industrial IoT, a large amount of data is being generated in research and development processes in the manufacturing domain. These include manufacturing procedures, enterprise management, and product transactions [8]. Data directly related to production efficiency are managed closely and respond to change. Among the master data in production, the yield and tact time are the two key metrics used by manufacturers to measure the production efficiency. Low yields indicate that manufacturers are producing defective products. This can cause low customer satisfaction, lost revenue, and increased production costs. Accurately estimating and managing tact time also enables the forecasting and planning of production volumes and production schedules. This, in turn, enables production to be planned efficiently and maintains the production line running smoothly. This also implies a reduction in the process lead time and thereby, a faster production and more rapid dispatch of products to customers [9]. An improvement in the efficiency of the production process can be an important indicator of customer satisfaction. This is because it enables producers to respond better to the diverse needs of their customers.

In this respect, both the datasets contribute to increased productivity and improved quality. As variable baseline information is critical for manufacturers to improve the production efficiency, reduce costs, and enhance customer satisfaction, manufacturers can monitor these data and undertake action to improve the overall business performance [10].

2.2. How to Estimate Manufacturer Master Data

Forecasting the data values of the two master datasets (yield and tact time) can help manufacturers optimize their production processes and ensure that they manufacture the products they require to satisfy the demand in a timely manner [11]. Several methods can be used to forecast data. One of these is to use historical data (where past performance data can be used to identify factors that can affect the yield and tact time) and predict the future values of these factors. Another method to predict the baseline information is to use a simulation. Simulations can be used to model the manufacturing process in a virtual environment. Moreover, the model can be used to test different operating conditions and determine their effect on the measured data [12].

This is a simplified way of saying that master data that must to be predicted are analyzed and applied to the manufacturing floor. Figure 1 shows the conventional methods of processing data by analyzing these using a structured system implementation logic based on past performance data. The processed data are ultimately applied based on the user's assessment, which is dependent on the user input.

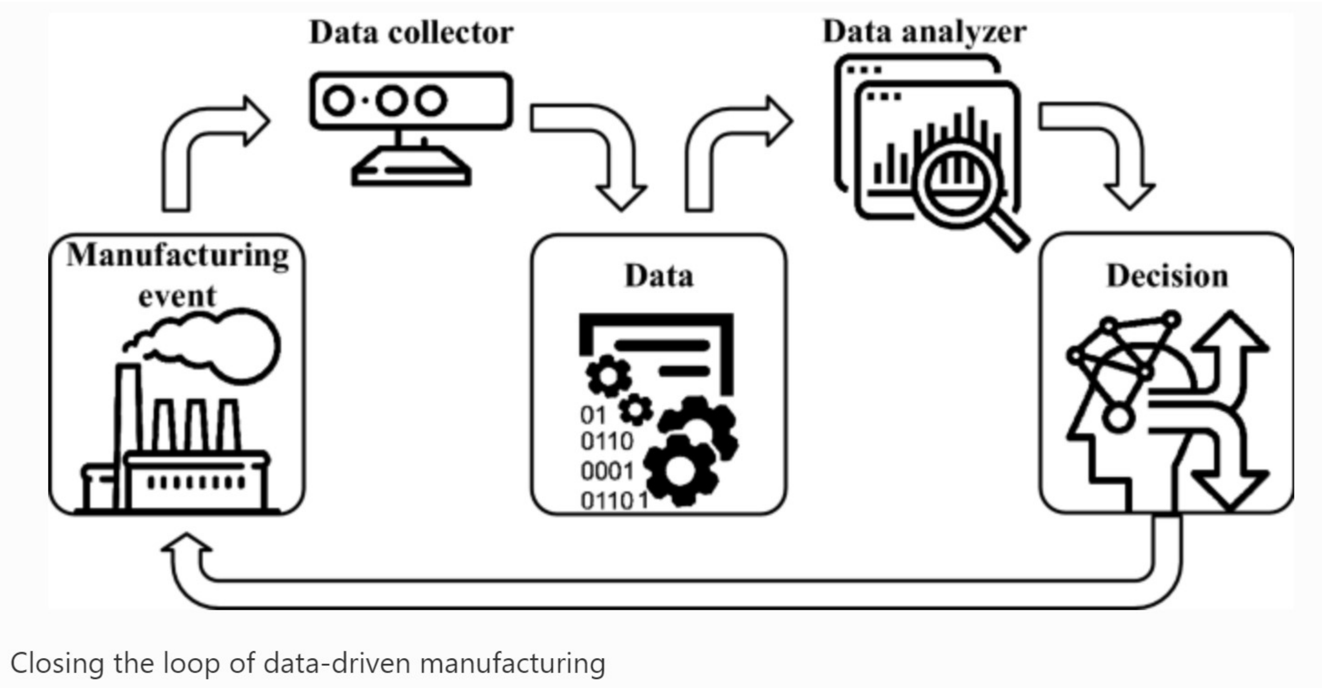


Figure 1. Traditional Method for Advanced Data Collection and Analysis in Data-Driven Manufacturing Processes [13].

2.3. Highly Advanced Methods to Manage Manufacturer Master Data

In a manufacturer's smart factory, the IoT enables sensors and other devices to collect and transmit data to monitor and control baseline information in real time. With rapid advances in AI technology and hardware performance, machine learning has attained a level of sophistication that manufacturers can use to identify patterns in historical data and predict baseline information [14]. This indicates that preprocessed master data can be analyzed using machine-learning algorithms to predict and reflect the results in real time [15]. Real-time data are limited by the fact that it does not fully account for the interactions and uncertainties between different variables. This can be overcome by utilizing various machine-learning algorithms. It is challenging for existing job-planning techniques to obtain effective optimal solutions using an individual numerical analysis method for complex distributed resource-planning problems. The effectiveness of the proposed hybrid methodology is demonstrated through a comprehensive case study of manufacturers [16]. To address uncertainty, many methods to predict data are being researched actively. These include the use of fuzzy logic to model ambiguous input variables and environmental factors and the use of nonlinear programming techniques to identify optimal decision variables [17]. Research and demonstrations are being conducted at present to construct machine-learning models based on real-time data collected from various sensors for identifying normal and abnormal conditions [18].

Taken together, these studies show that manufacturing master data management is evolving to focus on leveraging data, analytics, and automation. Adopting AI technology can optimize production processes, improve quality, manage reference information more efficiently and effectively, and improve the overall performance of the production process [19]. As shown in Figure 2, AI can help manufacturers collect and analyze data from sensors, machines, and other devices. This enables manufacturers to track the performance of their production processes and identify areas for improvement [20]. It can also be used to predict future data. It can help predict the future output, quality, and costs. This, in turn, can help manufacturers manage inventory levels and produce the products required to satisfy the demand. Overall, it can help manage the master's data more efficiently and effectively.

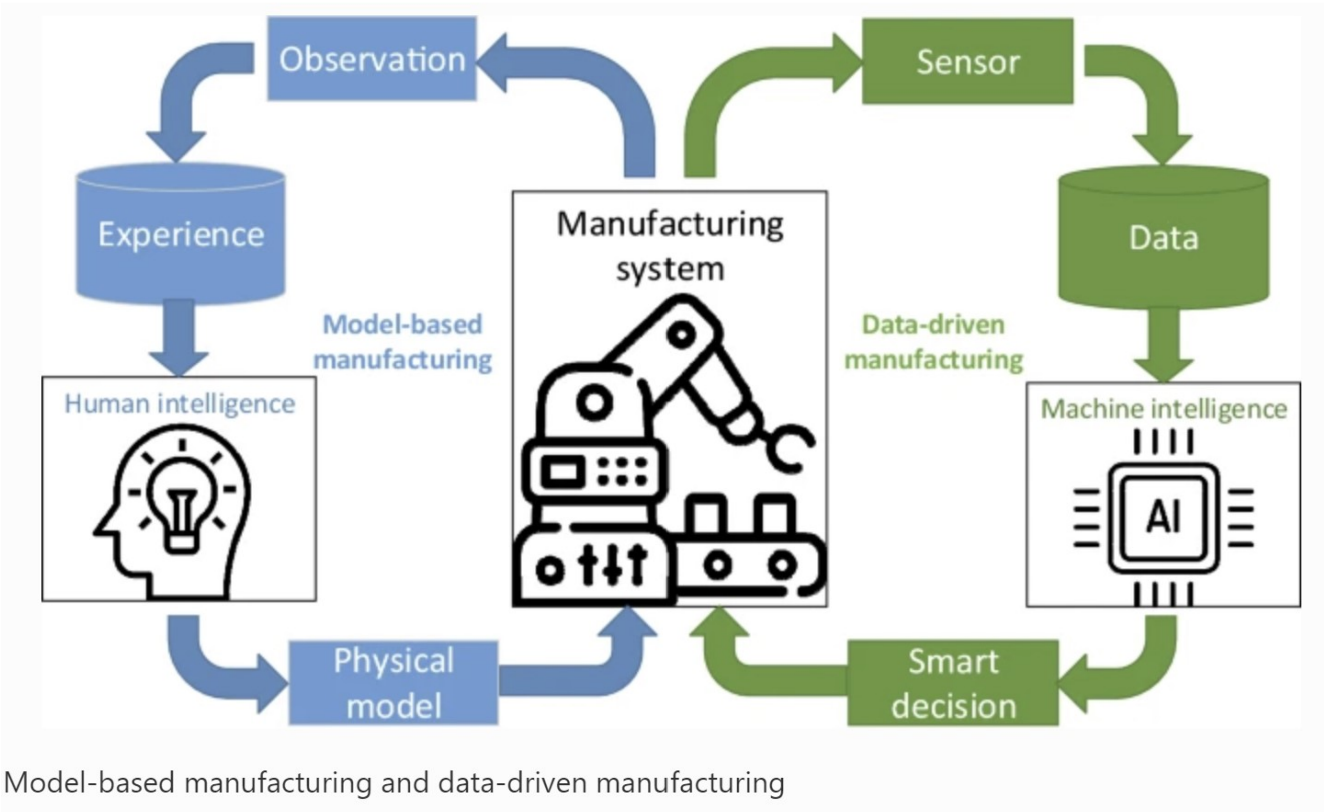


Figure 2. Advanced Data Collection and Analysis in Data-Driven Manufacturing Processes [13].

3. M5P Algorithms and Master Data in Production Planning System

In this study, the process of estimating the master data used by manufacturers and applying it to production planning is discussed. It describes how the master data are collected, preprocessed into a data form suitable for the process, and then measured using an algorithm to be applied. The study also describes the behavior of the machine-learning algorithm M5P applied in this study and proposes a data application process. The purpose is to apply the predicted values to a production planning system and compare these with existing results to provide effective indicators for production planning.

3.1. Forecasting Variable Master Data

Research has revealed that most manufacturers have experienced consequences related to additional production resources, extended lead times, reduced product quality, and low performance owing to data management issues. This indicates that data errors are common in business processes [21]. Master data management plays a highly important role in production planning. In the production process, baseline information such as the production volume and production time comprise the core information required to establish production plans and operate efficiently. Among the master data, tact time and yield mostly have constant values. However, these display inconsistent and unstable characteristics in an altered manufacturing environment, such as manual processes, newly developed processes, or the addition of new facilities [22]. Rather than manage the master data as user-managed baseline information, machine-learning algorithms are used to make predictions based on process data such as performance, to establish production plans [23]. Accurate forecast data can optimize the production schedule guidance to obtain the optimization of the total production time. Therefore, ensuring the proper use of existing resources to meet the basic requirements of the production schedule has important theoretical significance for the actual production of the enterprise [24]. The target master data to be predicted are defined as follows.

1. Tact Time: The time required to produce one product. In the production process, tact time is an important factor that determines the production speed of the production line. For efficient production, it is necessary to optimize the tact time of the production line and establish a production plan based on it. Keeping accurate and up-to-date information about tact time through master data management ensures that production plans are well-aligned with actual production.
2. Yield: Indicates how many of the products produced during the production process meet the quality standards. Yield directly affects product quality and production performance, so it is an important factor to consider when planning production.

Figure 3 shows that the existing IT system is implemented such that users master data information through the UI , whereas the new IT system collects and analyzes data through sensors located in resources other than the users. It predicts the current data as well as the future data to be used for planning. Manufacturing processes require flexibility and the capability to reconfigure themselves to address critical challenges. To achieve this, it is important to obtain relevant information in real time to make strategic decisions, optimally utilize available resources, and remain competitive in the market. A method to achieve this is to use machine-learning to predict and provide effective data. Machine-learning algorithms are utilized to improve production planning and address scheduling challenges [25]. It is challenging for conventional job-planning techniques to obtain effective optimal solutions to complex distributed resource-planning problems using individual numerical analysis methods [26]. Thus, machine-learning algorithms have been used to develop methods to improve production planning. In this study, the M5P algorithm is used in the API to predict the values of the two types of master data. The objective is to establish a production plan and compare it with the existing master data plan, and thereby help users finalize the production plan.

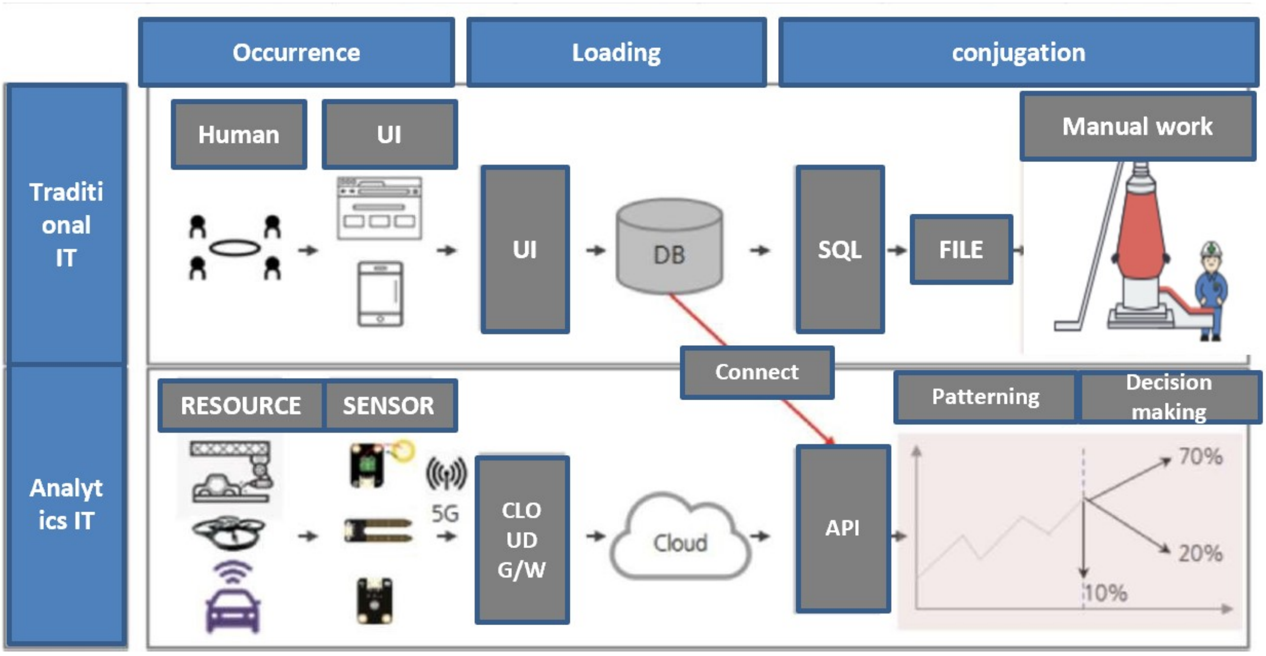


Figure 3. Traditional IT compared to Analytics IT.

Artificial Intelligence (AI) is an emerging branch of data analysis, widely used when trying to obtain intrinsic relationships between data [27]. This is a new way of analyzing data to understand relationships. It utilizes machine learning algorithms as a prediction method. Machine learning algorithms can be broadly categorized into supervised learning, unsupervised learning, and reinforcement learning [19]. Supervised learning is a method that uses input data and data with correct answers (labels) to train a model. Using the training data, a predictive model is generated that could predict the correct output for a new input. The predictive model learns the relationship between a given input and output to generate a decision boundary [28]. A typical flowchart of a supervised learning algorithm in machine learning is shown in Figure 4. Based on this flowchart, here is how the study was conducted.

1. Data collection: The input data required for training and the correct answers (labels) are collected. The data required for training for both yield and tact times are the master data consisting of existing baseline information. The correct answer data are calculated based on the performance.
2. Data preprocessing: The collected data are processed into an analyzable form, and the necessary preprocessing tasks are performed. These methods include data cleaning, attribute scaling, and outlier handling. The statistical program R is used to organize the horizontal database data into vertical data. This is the means by which the features are represented as columns in a model, wherein each row represents a data point. When converted into vertical data, the features are placed in columns corresponding to the model input matrix. This makes it suitable for the model to process the data. The obtained yield and tact-time training data are organized into vertical data by type. In machine learning, the processing of vertical data has the advantage of standardizing the data structure. This facilitates their application to various models and libraries.
3. Select/extract attributes: The attributes required for training are selected or extracted. In this step, one can analyze the characteristics of the data and select important features or extract new features. In this study, information such as ITEM and RESOURCE is extracted from the machine learning calculation because these are not directly related to the target data calculation. Moreover, the data to be included in the calculation can be selected.
4. Select a model: The best model for the given problem is selected. The M5P algorithm provided by Weka (a Java machine learning library) is used in this study.

5. Train the model: The selected model is trained using the training data. During the training phase, the model learns the relationship between the input data and correct answers (labels) for that data so that it can make predictions.
6. Evaluate the model: The performance of the trained model is evaluated. This is accomplished by examining the model's prediction results using test data and calculating the evaluation metrics. The main evaluation metrics are the accuracy, precision.
7. Apply the model: The trained model is applied to new data to make predictions. The results of the predictions for new data are applied to accomplish the objective. In this study, the data predicted by machine learning are incorporated into the master data to generate a new production plan.

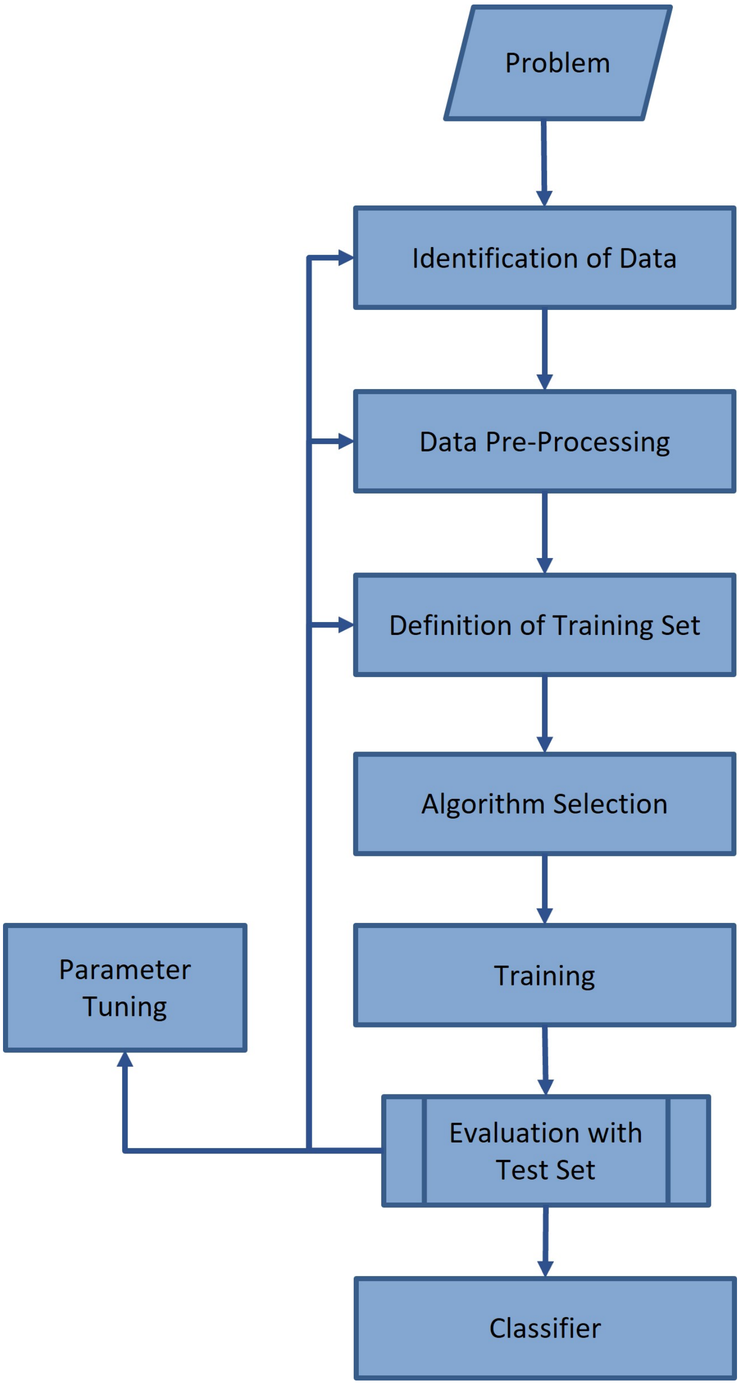


Figure 4. Machine Learning Supervise Process [29].

This study calculates two target master datasets by analyzing the performance information and predicting their values using a supervised learning algorithm, in conjunction with their corresponding target values.

3.2. M5P Algorithm

The M5P algorithm has the following features and benefits. It has advantages in handling categorical features (categorical variables). The algorithm includes a way to handle categorical variables internally, so they can be directly included in the predictive model without the need to preprocess the data. This simplifies the model development process and saves time. It also allows the user to automatically determine appropriate bifurcation points to account for different levels of categorical variables to achieve optimal predictive performance. The M5P algorithm typically has a faster execution speed than other complex regression algorithms. It is useful for small-sized datasets or in situations with time constraints. Applying the M5P algorithm also allows the relationship between data and rules to be described and analyzed, to predict the numerical characteristics of the target variable [30]. One of the main advantages of model trees is that these can efficiently handle a large number of datasets with many attributes and dimensions. These are also known to be robust when addressing missing data [31]. These two features are the main reasons that this algorithm was selected for this study. The capability to rapidly process large amounts of data with missing and volatile information, and achieve predictability is important in daily production planning. As shown in Figure 5, there are three major steps in developing an M5P tree: tree construction, tree pruning, and tree smoothing. The M5 tree construction process attempts.

1. Tree Construction: The M5P algorithm constructs a regression tree by recursively partitioning the training dataset based on the attribute values. The splitting process aims to find the attribute that provides the best split, often based on criteria such as information gain or variance reduction. The algorithm continues splitting until a stopping criterion is met, such as reaching a minimum number of instances per leaf or a maximum tree depth. The constructing process attempts to maximize a measure called the standard deviation reduction (SDR) [32].
   Equation (1) of SDR is shown as follows, where H is the instances dataset that stretch the node, H_i is the set that is received from a divided node according to a given attribute, and sd is the standard deviation of $\bar{H}$ [33].
$$SDR = sd(H) - \sum_i \frac{|H_i|}{|H|} \times sd(H_i),$$
$$sd(H) = \sqrt{\sum_1^N \frac{(H_i - \bar{H})^2}{N-1}}, \quad (1)$$
$$\bar{H} = \sum_i^N \frac{H_i}{N},$$
2. Model Pruning: Pruning is an optional step that aims to simplify the tree and reduce overfitting. Pruning techniques, such as subtree replacement or subtree raising, can be applied to remove unnecessary branches or nodes from the tree without significantly affecting the performance. Pruning helps to generalize the model and improve its predictive capabilities on unseen data.
3. Tree Smoothing: The M5P algorithm trains a linear regression model on the leaf nodes during the process of constructing the model tree. The predicted value at each leaf node is calculated based on the conditions at that leaf node. However, the model tree may be too complex or tend to overfit the data. To address these issues, tree smoothing is used. Tree smoothing is a way to increase the smoothness of predictions, and it involves combining the predictions of a leaf node with the predictions of its parent node, with some adjustments. This reduces the volatility of the predictions and allows

the complexity of the model tree to be controlled. By regression, the final value is smoothed by combining the current value with the predicted value from the linear regression as the following Equation (2).

$$\bar{T} = \frac{Nt + KA}{N + K} \quad (2)$$

where $\bar{T}$ is the predicted value shift to the higher level of the next node, N is the total number of training instances that shift to the next lower node, Nt is the predicted value shifted from the lower node to the present node, A is the predicted value by the node at this node, and K is a constant value [33].

In Figure 6, the M5P algorithm can perform both regression and decision trees. Pruning allows nodes to include linear models instead of constant values. The M5P algorithm is an algorithm for solving regression problems which is used to predict a continuous output value from given data [34]. Create a decision tree to make a classification, and then create a linear model for the nodes. The tree is constructed based on the features of the data, and prediction is performed by dividing the data according to the conditions. The leaf contains a linear model, which can express the linear relationship between the features of the data and the target [35].

The "Unpruned" option is set to false, which means that pruning is enabled. In general, pruning allows the model to be more concise and provide more generalized results. Pruning is a technique for reducing the complexity of a model by removing unnecessary branches from a tree model.

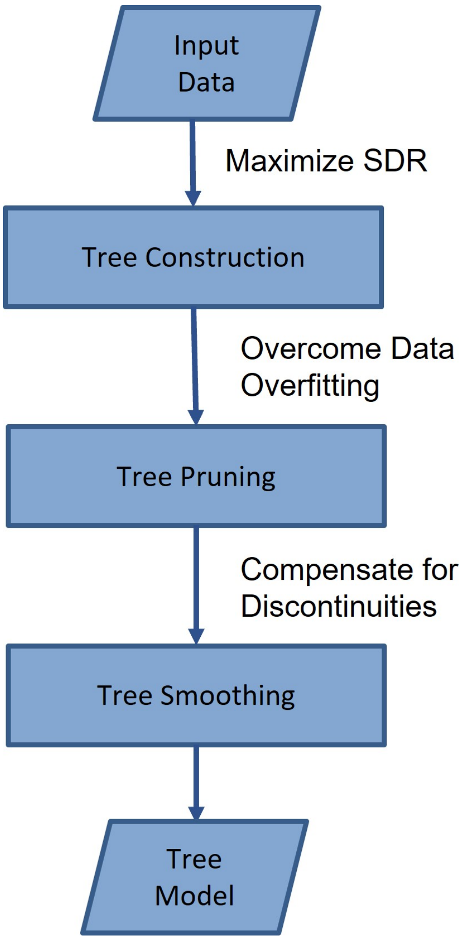


Figure 5. M5P Algorithm Training Process Flowchart.

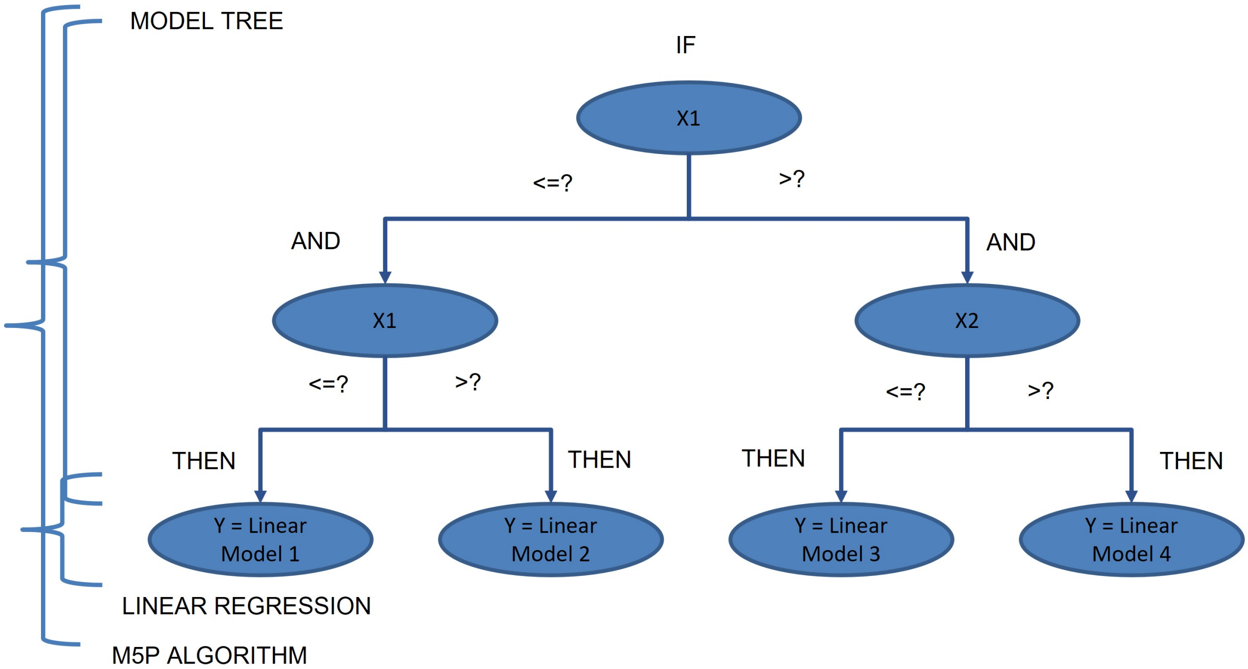


Figure 6. M5P tree algorithm.

3.2.1. Linear Regression

A linear regression algorithm is a regression analysis technique that models the linear correlation between a dependent variable (y) and at least one independent variable (x). It can be a simple linear regression based on an explanatory variable or a multiple linear regression based on more than one explanatory variable [36]. It is called simple linear regression when based on a single explanatory variable, and multiple linear regression when based on more than one explanatory variable. Linear regression uses a linear prediction function to model the regression equation, and the unknown parameters are estimated from the data. Equation (3) a formula is a way to model a linear relationship, and a method is a formula expressed in vector form. The formula for the linear regression algorithm, where y is the predicted value, x are the input variables, β are the weights for each variable, and b is the bias [37]. This form of a linear equation models a linear relationship between the input variables, where the weights represent the importance of each variable. The linear model uses the data to adjust the weights and bias as it is trained. Typically, methods such as Ordinary Least Squares or Maximum Likelihood Estimation are used to estimate the parameters of the model. Linear models are less affected by the scale of the attributes, so data preprocessing is relatively simple.

In this study, while generating a linear regression formula with tact time as the result of the M5P algorithm, the metadata included the RESOURCE ID and ITEM ID. Meanwhile, the feature was unspecified. Operation tact time, minimum tact time, etc. The target value was the real tact time. Based on these data, a linear regression formula was generated in the form of (3) to obtain the predicted value.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon \quad (3)$$

In this study, the results of the data applied to the production planning system were implemented through machine learning using a formula model (3) to predict the resulting data.

3.2.2. Decision Tree

When a decision tree is constructed, several branches may introduce variances in the training dataset owing to noise or outliers. This problem has been addressed as an overfitting in tree pruning, which uses statistical procedures to eliminate less accurate branches. It generally includes pre- and post-pruning [18]. Decision trees are among the most popular supervised learning algorithms used in machine learning. It learns decision rules based on the values of the attributes in the data and represents these in a tree structure [38]. The M5P algorithm is a type of decision-tree algorithm. It is a part of an algorithm called model trees, which works in a manner that is moderately different from that of decision trees. Unlike decision trees, the M5P algorithm trains and uses linear models at the leaf nodes. The hyperparameters of the M5P algorithm tree used by Weka are as follows:

1. Unpruned: This is a Boolean value that prevents pruning. The default value is false, indicating that pruning would be performed.
2. UseUnsmoothed: This is a Boolean value that prevents smoothing. The default value is false, which implies that smoothing is disabled.
3. MinimumNumberInstances: An integer value that specifies the minimum number of instances required on each node. The default value is 6. Setting it to a smaller value can increase the complexity of the model. In the M5P algorithm, "Minimum Number of Instances" is important hyperparameters used to control the complexity of the model [39]. These can be adjusted to improve the performance of the model.
4. BuildRegressionTree: This parameter is used when generating the regression tree and is set to a boolean value. The default value is true.
5. Max saveInstanceData: This parameter is a boolean value that determines whether to save the training data. The default value is false, and if set to true, training data are saved.

Decision Tree is used as a classifier for the M5P algorithm used in this study. The decision tree provides a structure for pruning unnecessary data and resolving overfitting based on the characteristics of metadata, such as ITEM ID and RESOURCE ID.

3.3. Production Plan System

Prior to the concept of SCM, companies connected various processes such as production, logistics, and accounting. Production, logistics, finance, accounting, and so on share information and resources through ERP systems [40]. However, as the supply chain complexity increases, a new system for managing the planning process becomes necessary. As a system to manage production planning, it was introduced with the characteristics of an advanced planning and scheduling (APS) hierarchy. The APS system is used to develop a production plan based on these two characteristics and existing data. The supply chain can function as a solution to multiple planning problems by optimizing goals and constraints through a model definition of the plan [41]. The master data (which contain basic information related to products, inventory, production facilities, supply chains, etc.) play a highly important role in production planning and management. This directly affects the accuracy and efficiency of production planning. Maintaining accurate and updated data and formulating production plans based on these improves the accuracy and flexibility of production schedules and enables effective inventory management and efficient supply chain operations.

In this study, a production plan was established using the APS system by reflecting the master data predicted by machine-learning results. A comparison of the established plan with the existing plan provided indicators that can be used as a reference for comparing results that affect the availability of facilities, such as the input quantity.

3.4. Process Implementation Tools

The data for this study were obtained in the CSV format and preprocessed using the statistical program R. The machine-learning models were generated using Weka.

3.4.1. Weka

In this study, Weka was used to apply the machine-learning algorithms. This is an open-source platform for machine learning and data mining. It is used to develop and test machine-learning algorithms. It can perform various machine-learning tasks such as data preprocessing, classification, regression, and clustering [42]. Weka is written in Java and provides libraries that implement various machine-learning algorithms. It can be used to analyze data and construct predictive models. The M5P algorithm described was implemented in the library. The library allows for the application of machine-learning algorithms.

3.4.2. R

R is the programming language used for data analysis and statistical modeling. It is open-source software that provides users with the flexibility to perform a variety of sophisticated analyses. The statistical program R provides several advantages in machine learning. It provides a wide range of statistical techniques and functions. R is highly active in the machine learning and data science communities. Moreover, there is a community of users who share knowledge, information, and support for problem solving [43].

Given these advantages, R was used to preprocess the collected horizontal data. Additionally, a new data attribute was generated. It enabled us to exclude unnecessary data and group these. It can be plugged into Weka to use most features of R for data preprocessing.

3.5. Advanced Production Planning Methods for Manufacturing Processes

Production planning is an important step in planning and coordinating the process by which an organization produces a product or service. To increase the predictive power of the master data, we experimented with new technologies and approaches to produce more accurate and efficient results.

This study utilized a data-driven predictive model to forecast production requirements. Yield and tact-time data were used to generate predictions using machine-learning-based algorithms. Data were collected and analyzed. Predictive models can be used to forecast data, which can then be used to plan production. Based on the data collected, an APS system was used to establish a production plan. Various scenarios can be simulated during production planning, and an optimal production plan can be derived.

4. Experimental Settings up

4.1. Design and Implement Production Planning Changes

Production planning is the process of determining the resources and work schedules required to produce a product or service. In this study, production planning followed the flow shown in Figure 7. A production plan was established based on the data required for planning. This plan was based on the results of the existing interval planning method and the change dataset predicted by machine learning. This study proposed a process that provides an indicator for comparing two planning results and specifies a more reasonable plan. The master data, based on the flowchart in Figure 7 are obtained through PL/SQL and changed into a CSV file. Before applying the machine learning algorithm, the data were preprocessed using R. This involved classifying metadata, features, attributes, targets, etc., and separating the data into training and test sets. Planning through machine learning predictions was established by utilizing a production planning solution. Create a production plan that results in the database and compare the plans using PL/SQL.

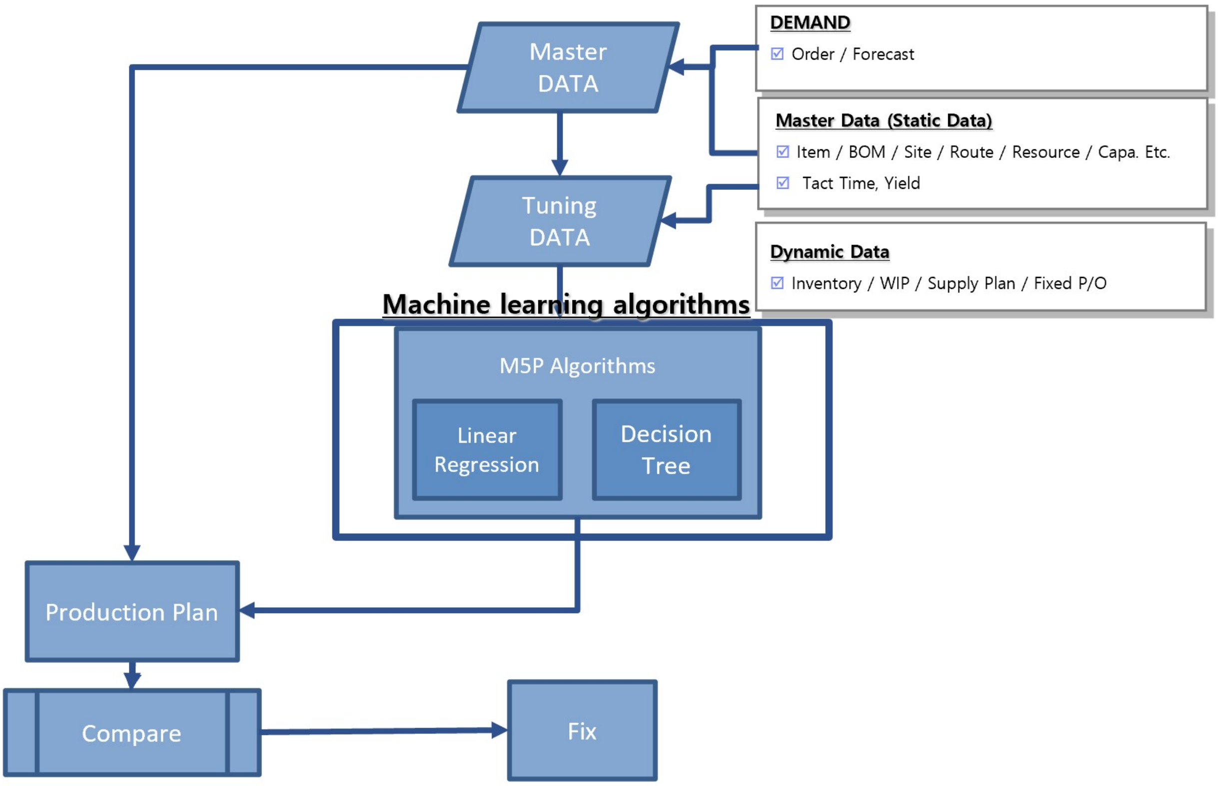


Figure 7. Machine Learning Reflection Comparison Production Planning Process.

4.2. Development Environment

The machine learning model implementation environment works on a laptop. The hardware environment for machine learning is shown in the table below (Table 1).

Table 1. Hardware Environment.

Hardware	Performance
CPU	13th Gen Intel(R) Core(TM) i7-1360P 2.20 GHz
RAM	16 GB
GPU	Intel® Iris® Xe Graphics

The environment in which the machine learning model was developed is shown in the table below (Table 2).

Table 2. Development Environment.

Type	
OS	Windows 11 Home 22H2
DBMS	Oracle 21c XE
Development Languages	Java 1.8
Development Tools	R 4.3.0, Weka 3.8.6

The programs implemented in Master Data Machine Learning Predictions were implemented in Weka.

4.3. Implementing a Machine Learning Model

Figure 8 is a flow chart of a machine learning application implemented using Weka. In the flow chart, check the order of preprocessing, training data, test data, classification, applying machine learning algorithms, and applying classifiers.

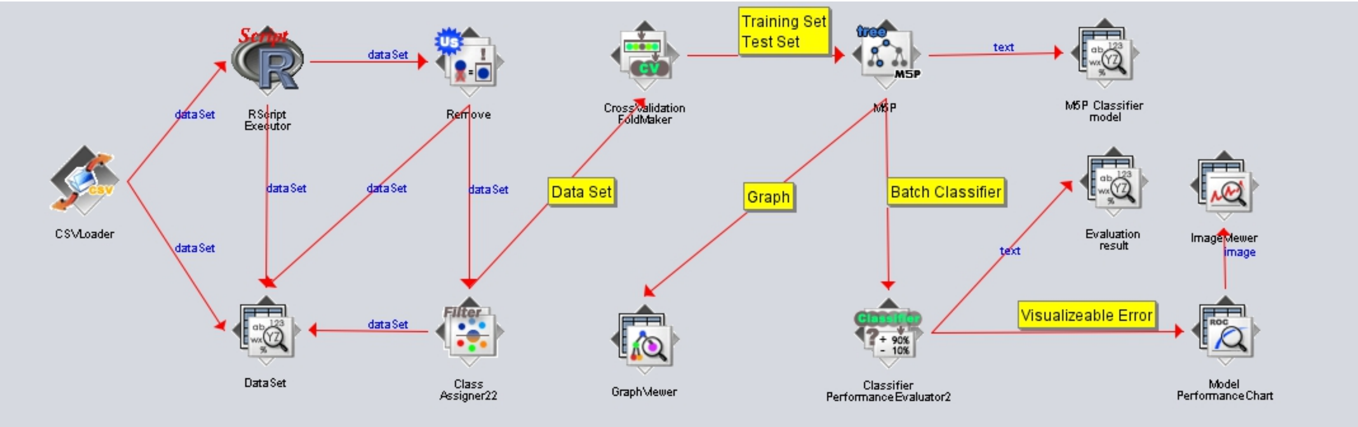


Figure 8. Machine-learning Model Flow by Weka.

The input data consist of master data required for production planning. Among the many master data, yield and tact time data are specified in this study. We extract the corresponding data from the manufacturer's database using PL/SQL. The input data were received as a CSV file and preprocessed using the R program. Raw data are horizontal data like Figure 9. The data are preprocessed to categorize each data and remove unnecessary data. The character data are categorized as metadata, and the variables that affect the prediction value are set as features. Based on the table data, the tact time series data are configured as a feature. We specify the prediction target variable and change it to a new dataset that is easier to apply to the machine learning algorithm. The pre-processed data were subjected to cross-validation fold maker. Cross-validation is used to evaluate the performance of a model by dividing a given dataset into multiple subsets, or folds. K-Fold Cross Validation was used, and the value of K was arbitrarily set to a constant. K means the number of folds to be divided. An experimental approach was needed to determine the value of K in our environment. The maximum K value is set to 10, and after experimenting with different values, we settled on 10 as the optimal value. After this process, the preprocessed data were divided into a test set and a training Set and used as input data. We set the training set ratio to 80% and the test data ratio to 20%.

In the WEKA environment, the M5P algorithm hyperparameters were tested based on the following four parameters. -N: Unpruned, -U: UseUnsmoothed, -R: BuildRegression-Tree, -M: MinimumNumberInstances.

The performance of the model can be controlled and optimized by adjusting these hyperparameters. The optimal values of the hyperparameters were tested based on the performance metric relative absolute error (RAE). It can be interpreted as follows: the smaller RAE is, the closer is the prediction to the true value. Table 3 shows the results, and the hyperparameter settings of Unpruned, UseUnsmoothed, and MinimumNumberInstances 6 show the best predictive metrics. Testing the hyperparameter settings defined the parameter values for this experiment.

Table 3. RAE metrics by Hyperparameter.

RAE (%)		-M 4.0	-M 5.0	-M 6.0	-M 7.0	-M 8.0
	NONE	10.042	10.0494	10.1141	10.5581	10.7338
-R	NONE	16.855	16.7616	16.764	16.7879	16.7879
	-N	16.3638	16.3637	16.3632	16.3665	16.3676
	-U	10.8224	10.7141	10.733	10.9508	10.9508
-N	NONE	9.5442	9.5456	9.5935	9.8459	9.9783
	-U	8.8193	8.7659	8.7515	9.0278	9.1206
-U	NONE	9.2306	9.2141	9.2728	9.9319	10.1855

DATE	RESOURCE_ID	ITEM_ID	OPERATION_TIME	TOTAL_WORKING_TIME	OPERATION_QTY	COMPLETE_QTY	REAL_TACT	OPERATION_TIME_TACT	MIN_TACT_TIME	AVG_TACT_TIME	MAX_TACT_TIME
2021-06-08	RMAI017D	A11568	12,680	9,800	3,000	1,000	9.8	12.68	9.8	10.09	9.8
2021-06-08	RMAI017D	A11568	10,400	9,800	3,000	1,000	9.8	10.4	9.8	10.09	9.8
2021-06-08	RMAI017D	A11568	16,700	9,800	3,000	1,000	9.8	16.7	9.8	10.09	9.8
2021-06-08	RMAI017D	A11568	13,280	9,800	3,000	1,000	9.8	13.28	9.8	10.09	9.8
2021-06-09	RMAI017D	A11568	10,400	9,800	3,000	1,000	9.8	10.4	9.8	10.09	9.8
2021-06-09	RMAI017D	A11568	3,136	3,136	960	320	9.8	9.8	9.8	10.09	9.8
2021-06-09	RMAI017D	A11569	1,372	1,372	420	140	9.8	9.8	9.8	10.09	9.8
2021-06-09	RMAI017D	A09649	19,160	15,680	4,800	1,600	9.8	11.975	9.8	10.06	9.8
2021-06-09	RMAI017D	A09649	16,280	15,680	4,800	1,600	9.8	10.175	9.8	10.06	9.8
2021-06-09	RMAI017D	A09649	26,060	15,680	4,800	1,600	9.8	16.2875	9.8	10.06	9.8
2021-06-10	RMAI017D	A09649	16,280	15,680	4,800	1,600	9.8	10.175	9.8	10.06	9.8
2021-06-10	RMAI017D	A09655	8,168	1,568	480	160	9.8	51.05	9.8	10.09	9.8
2021-06-10	RMAI017D	A09634	980	980	300	100	9.8	9.8	9.8	10.09	9.8
2021-06-10	RMAI017D	A11292	725	725	150	50	14.5	14.5	14.5	14.59	14.5
2021-06-10	RMAI017D	A11298	870	870	180	60	14.5	14.5	14.5	14.59	14.5
2021-06-10	RMAI017D	A11342	725	725	150	50	14.5	14.5	14.5	14.5	14.5
2021-06-10	RMAI017D	A11282	1,015	1,015	210	70	14.5	14.5	14.5	14.59	14.5
2021-06-10	RMAI017D	A09516	19,160	15,680	4,800	1,600	9.8	11.975	9.8	10.09	9.8
2021-06-10	RMAI017D	A09516	23,180	15,680	4,800	1,600	9.8	14.4875	9.8	10.09	9.8
2021-06-10	RMAI017D	A09516	19,160	15,680	4,800	1,600	9.8	11.975	9.8	10.09	9.8

Figure 9. Example of Tact time Data Preprocessed with PL/SQL.

When the data are trained with the M5P machine learning model, they are classified through a decision tree, and a linear regression formula is generated at the node.

The pseudocode for the M5P algorithm in the Weka Library is shown below (Algorithm 1).

The M5P algorithm is a decision-tree-based classifier. A decision tree is an algorithm that uses a tree structure to classify data. The M5P algorithm is a variation of a decision tree, which is a good model for regression problems. The M5P algorithm establishes criteria for partitioning the data and then performs another partition in each partitioned region to form a tree. This process means generating rules for classifying and predicting data. The result of the algorithm for deriving production plan results becomes the forecast input data to be applied to the production plan.

4.4. Configure Machine Learning Model Training Input Data

Data form the foundation for all machine-learning models. For a machine-learning model to learn, clean data samples should be fed continuously into the system during training. The desired task may not be achievable if the collected data are highly imbalanced or inadequate. To overcome this problem, performance-based data (excluding human-predicted data) were collected. The tact time and yield were calculated based on the production line, work start time, end time, input quantity, finished quantity, and so on for each standard production model as the input data.

In Figure 9, the yield and tact-time information data are preprocessed in PL/SQL and generated as a CSV file to be set as the default input data. The training dataset comprised over 10,000 cases. The data show that the average tact time, minimum tact time, maximum tact time, and actual performance-based tact time differed. The problem is to select the most appropriate value from the data to be applied to the plan. It is necessary to experiment with the data to generate predictions.

Algorithm 1 M5P Algorithm.

```
function BUILDM5P(dataset)
    Create a new node
    if all samples in the dataset have the same output value then
        return the node with the corresponding output value
    end if
    if there are no more features left to split on or stopping criteria are met then
        return the node with the average output value in the dataset
    end if
    Select the best attribute to split the dataset
    Assign the selected attribute to the current node
    for each possible value of the selected attribute do
        Partition the dataset into subsets based on the value of the selected attribute
        if a subset is empty then
            Create a leaf node with the average output value in the parent dataset
            Assign the leaf node as the child of the current node
        else
            Recursively call BUILDM5P on the subset
            Assign the returned subtree as the child of the current node
        end if
    end for
    return the constructed M5P tree
end function
procedure MAIN
    Load training dataset X and target values y
    Call LINEARREGRESSION function with X and y as input
    Get the trained weights w and bias b
    Use the trained model to predict target values for new instances
end procedure
```

Two results were obtained after the machine-learning model was trained using the M5P algorithm. A decision tree divides input data into features, determines the best decision rule for each partition, and represents it as a tree-like structure. This enables data classification or prediction. Decision trees are intuitive and convenient-to-interpret models. Techniques such as pruning have been used to prevent overfitting. The result of a Decision Tree model is a structure like the one shown in Figure 10.

Each node in the Decision Tree sorts through the metadata and prunes unnecessary outliers to create the linear model at the last node.

4.5. Production Planning Process with M5P Algorithm

The pruned tree shows that the decision-tree algorithm was applied and that the nodes were organized in a linear model. A linear model is generated as many times as there are nodes in the decision tree and generates multiple formulas. In the above formula, the final value is the vertical set of data from the preprocessed input data in Figure 11.

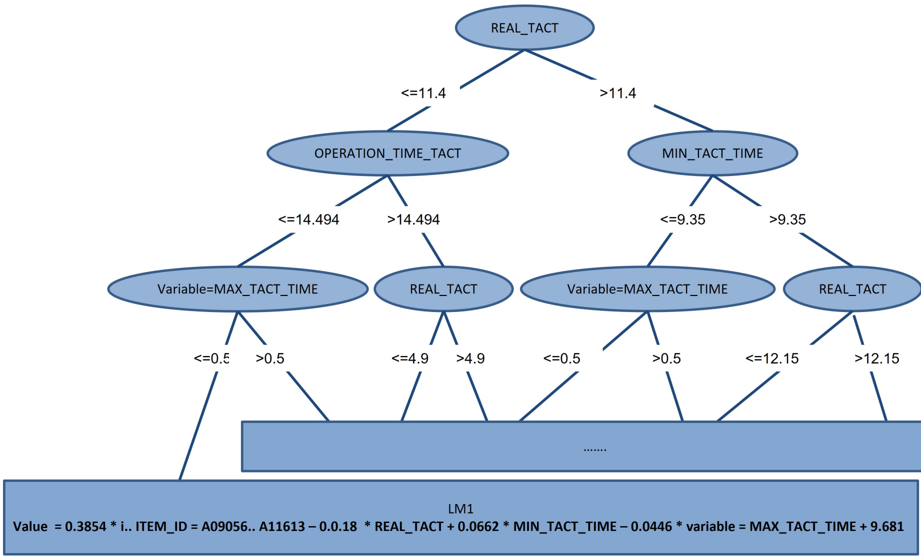


Figure 10. M5 unpruned model tree generated by the M5P algorithm using Weka.

```
Scheme:    M5P
Relation: R-data-frame-weka.filters.unsupervised.attribute.ClassAssigner-C6

M5 unpruned model tree:
(using smoothed linear models)

REAL_TACT <= 11.4 :
|   OPERATION_TIME_TACT <= 14.494 :
|   |   variable=MAX_TACT_TIME <= 0.5 : LM1 (21/0.557%)
|   |   variable=MAX_TACT_TIME >  0.5 : LM2 (22/0%)
|   OPERATION_TIME_TACT >  14.494 :
|   |   REAL_TACT <= 4.9 : LM3 (4/6.104%)
|   |   REAL_TACT >  4.9 :
|   |   |   variable=MAX_TACT_TIME <= 0.5 : LM4 (5/0.637%)
|   |   |   variable=MAX_TACT_TIME >  0.5 : LM5 (5/0%)
REAL_TACT >  11.4 :
|   MIN_TACT_TIME <= 9.35 :
|   |   variable=MAX_TACT_TIME <= 0.5 :
|   |   |   MIN_TACT_TIME <= 7.24 : LM6 (8/0%)
|   |   |   MIN_TACT_TIME >  7.24 : LM7 (21/0.467%)
|   |   variable=MAX_TACT_TIME >  0.5 : LM8 (31/0%)
|   MIN_TACT_TIME >  9.35 :
|   |   MIN_TACT_TIME <= 12.15 :
|   |   |   variable=MAX_TACT_TIME <= 0.5 :
|   |   |   |   REAL_TACT <= 14 : LM9 (4/0%)
|   |   |   |   REAL_TACT >  14 :
|   |   |   |   |   OPERATION_TIME_TACT <= 16 : LM10 (4/0%)
|   |   |   |   |   OPERATION_TIME_TACT >  16 :
|   |   |   |   |   |   OPERATION_TIME_TACT <= 20.561 : LM11 (3/0%)
|   |   |   |   |   |   OPERATION_TIME_TACT >  20.561 :
|   |   |   |   |   |   |   OPERATION_TIME_TACT <= 21.807 : LM12 (3/8.507%)
|   |   |   |   |   |   |   OPERATION_TIME_TACT >  21.807 : LM13 (3/8.507%)
|   |   |   variable=MAX_TACT_TIME >  0.5 : LM14 (16/0%)
|   |   MIN_TACT_TIME >  12.15 :
|   |   |   i..ITEM_ID=A09506,A09858,A10091,A10142,A11617,A11620,A11614,A11611,A10143,A10139,A11613 <= 0.5 : LM15 (9/2.252%)
|   |   |   i..ITEM_ID=A09506,A09858,A10091,A10142,A11617,A11620,A11614,A11611,A10143,A10139,A11613 >  0.5 :
|   |   |   |   variable=MAX_TACT_TIME <= 0.5 : LM16 (4/0%)
|   |   |   |   variable=MAX_TACT_TIME >  0.5 : LM17 (3/0%)

LM num: 1
value =
        0.3855 * i..ITEM_ID=A09506,A09858,A10091,A10142,A11617,A11620,A11614,A11611,A10143,A10139,A11613
        - 0.0183 * REAL_TACT
        + 0.0662 * MIN_TACT_TIME
        - 0.0446 * variable=MAX_TACT_TIME
        + 9.681
```

Figure 11. Linear model created with the M5P algorithm using Weka.

Through the GUI of the Weka Library, you can check the graphs and metrics of the M5P algorithm results for each equipment and model. Objects with the prefix "predicted" have a Y-value, indicating that they are the result of that machine learning. An example of one of the resulting values can be seen in the Figure 12. A model performance chart is used to evaluate the prediction accuracy and performance of the model by visualizing the relationship between predicted and actual values. The predicted value is called "predicte-dREAL TACT" and the actual value is called "REAL TACT". The chart shows how well the predicted and actual values match. The difference between the predicted and actual values is visualized.

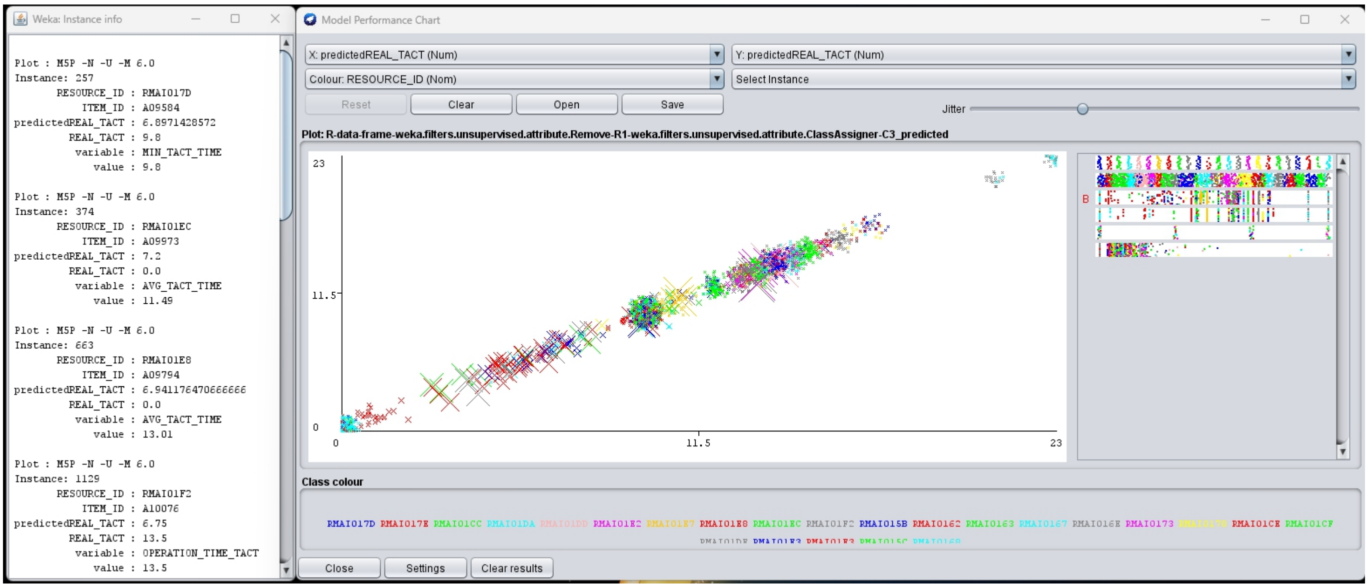


Figure 12. M5P algorithm Results predicted by Machine learning.

In this study, the target data for prediction were Real tact time and the Y value is Predicted real tact time as a result of the algorithm. All metadata are composed of different colors and can be classified in the graph, the composition data of the result value can be checked.

Various metrics have been used to evaluate the performance of machine-learning regression models, including the mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R-squared). MSE is a metric that squares the difference between the predicted value and the actual value, then calculates the average. The smaller the MSE, the smaller the model's prediction error. MAE is a metric that calculates the average of the absolute differences between predicted and actual values. The smaller the MAE, the smaller the model's prediction error. R-squared is a measure of how well a model explains the given data. It represents the percentage of the total variability in the given data that can be explained by the model. The coefficient of determination has a value from 0 to 1, with values closer to 1 indicating that the model explains the data well. These metrics are used to evaluate and compare the performance of the regression model in various aspects. In Figure 13, the evaluation metrics for the performance of the regression model can be seen. Shows the results of machine learning performance metrics with the optimal values of hyperparameter settings -N, -U, and -M 6.0 applied to the test.

```
=== Evaluation result ===

Scheme: M5P
Options: -N -U -M 6.0
Relation: R-data-frame-weka.filters.unsupervised.attribute.Remove-R1


Correlation coefficient                  0.9598
Mean absolute error                      0.2791
Root mean squared error                  1.26
Relative absolute error                  8.7515 %
Root relative squared error             28.0905 %
```

Figure 13. M5P Algorithm Machine Learning Results.

Like tact time, yield can also be predicted using the M5P algorithm, and the resulting value, M5P Linear Regression, is reflected as new master data. Input the results of machine learning predictions into a database so that they can be used as supplementary indicators for existing planning results.

5. Results and Discussion

5.1. Experiment Results

This study aimed to (1) predict master data that can potentially fluctuate (rather than being constants) using machine-learning algorithms and (2) present the results of applying those predictions. The M5P machine-learning algorithm predicts the yield and tact time data that may fluctuate among the master data, and establishes a production plan that reflects these values. This provides auxiliary indicators for production management. Tact time and yield are absolute indicators of work time and quantity. These may not have exact values depending on the process characteristics.

Figure 14 shows that an error exists between the actual plan and the tact time predicted using the M5P algorithm.

A production plan with a large error in the actual production entity can adversely affect many production-related indicators such as staffing and equipment utilization. An increase of the number of inputs requires additional raw materials, labor, and equipment. An increase in the number of inputs increases the inventory of produced products. This can increase the unit cost of production, decrease competitiveness, and reduce the flexibility of the production process. Therefore, it is important to consider these issues and plan production when inputs increase. Determining the appropriate input amount and balancing productivity gains with cost efficiency are critical issues. Figure 15 shows that the input quantity has increased significantly compared to the original plan due to the predicted yield, due to the increase in input quantity, changes in inventory management, and facility utilization rate, etc. This results in large differences when comparing detailed planning results. Similarly, Figure 16 shows that the working time per parking lot increased. When planning for a modified baseline production, the differences in yield and tact time can indicate a difference in the overall manufacturing plan. Fluctuations are important indicators that should be assessed during production management. Even a small difference in yield or tact time can have a significant impact on the line balance as the production volumes increase. It can also generate a series of supply chain issues such as inventory management and due dates. Therefore, a new plan should be formulated to reflect the erroneous results.

WORK_ORDER_ID	RESOURCE_ID	ITEM_ID	RESOURCE_ID	START_TIME	END_TIME	ML_MODEL_TACT	WORKING_TIME_DIFF
011001511630^1	RMAI0164	A09576	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001511638^1	RMAI0164	A09576	RMAI0164	2021-05-25	2021-05-25	9.8	-600
011001511906^1	RMAI0164	A09576	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001518442^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	-2880
011001513132^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	2^{-36}
011001511735^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001518450^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.799043062	3^{-36}
011001513074^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	-600
011001513242^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001513391^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.827586207	0
011001513559^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.778761062	3^{-36}
011001513827^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.804878049	0
011001518579^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001518923^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.795811518	-600
011001514589^1	RMAI0164	A09507	RMAI0164	2021-05-25	2021-05-25	9.803827751	-6^{-36}
011001511777^1	RMAI0164	A09508	RMAI0164	2021-05-25	2021-05-25	9.8	8^{-36}
011001511548^1	RMAI0164	A09508	RMAI0164	2021-05-25	2021-05-26	9.8	-2880
011001507031^1	RMAI0164	A09508	RMAI0164	2021-05-26	2021-05-26	9.8	8^{-36}
011001511198^1	RMAI0164	A09508	RMAI0164	2021-05-26	2021-05-26	9.8	-600
011001511440^1	RMAI0164	A09508	RMAI0164	2021-05-26	2021-05-26	9.8	8^{-36}
011001512042^1	RMAI0164	A09508	RMAI0164	2021-05-26	2021-05-26	9.8	0
011001558747^1	RMAI016F	A11042	RMAI016F	2021-05-27	2021-05-27	9.8	8^{-36}
011001559063^1	RMAI016F	A11042	RMAI016F	2021-05-27	2021-05-27	9.8	8^{-36}

Figure 14. M5P Algorithm in Learning reduces learning Tact time Compared to Original Data.

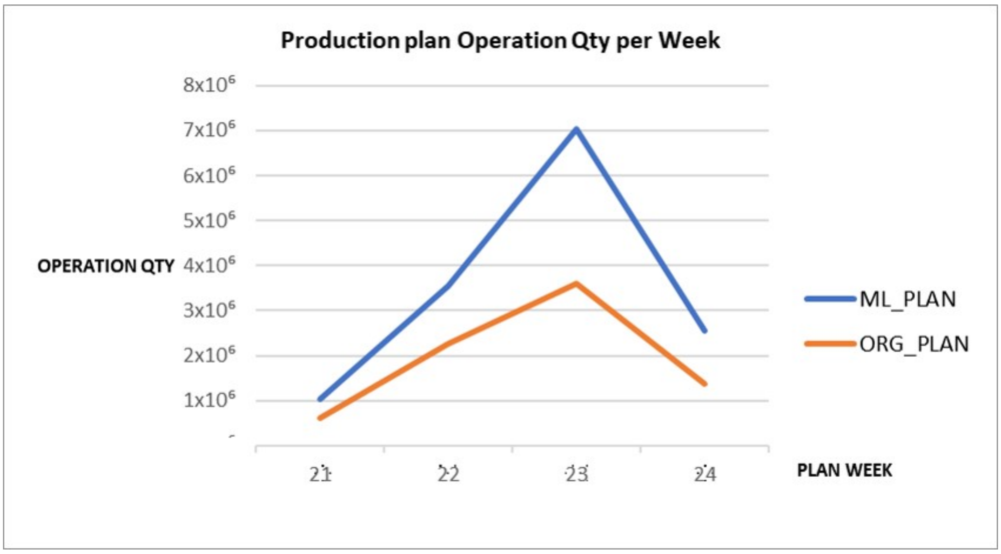


Figure 15. Compare the Quantities of the Existing Plan and The Machine Learning predictive Model Plan.

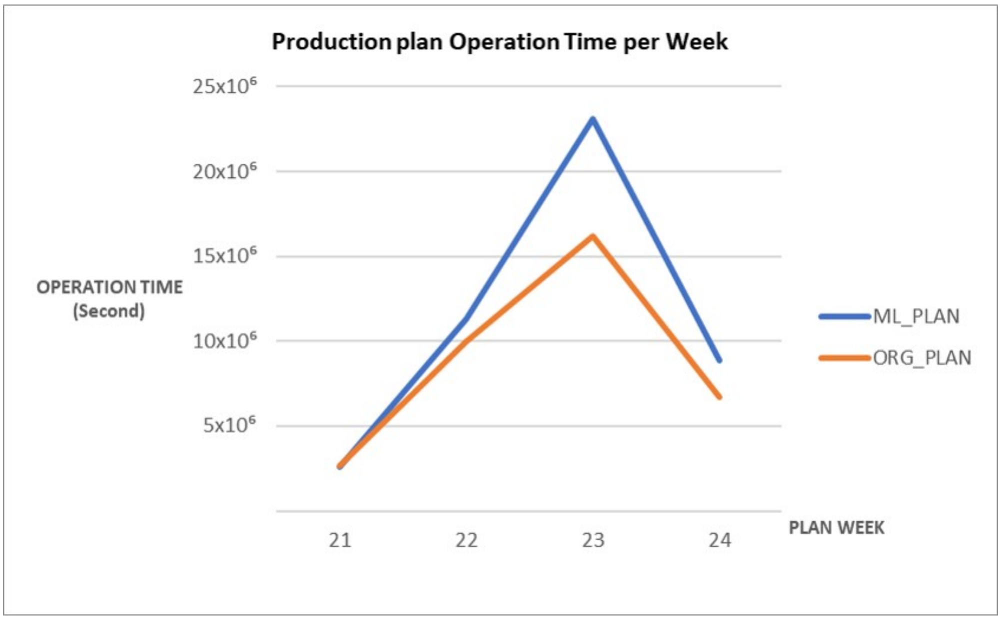


Figure 16. Compare the Operation times of Traditional Planning and Machine Learning Predictive Model Planning.

Increasing the input quantities and working hours are important issues in production management. If these fluctuations can be predicted and considered in an actual production plan, the accuracy of the schedule, efficiency of the production line, and management of resources can be better controlled. This would, in turn, enable production process optimization at an unprecedented level. Even when data is managed with existing baseline information, it would be feasible to predict the future and plan more flexibly if the baseline data can be predicted based on the existing performance, and to use it as a supplementary indicator.

5.2. Experiment Discussion

The M5P algorithm was applied in the experiment. It has the advantage of handling continuous variables, making it suitable for numerical data analysis. M5P has excellent model transparency by generating a model tree that can be easily interpreted and understood. On the other hand, the existing Decision Tree algorithm lacks the Hyperparameters available in traditional decision tree algorithms. M5P is primarily designed for continuous variables and may not handle categorical variables effectively. Also, without proper pruning or regularization techniques, M5P can be prone to overfitting, especially when the model tree becomes overly complex. If, as in this experiment, there are many features in the ITEM ID and RESOURCE ID, the size of the tree may increase, making the model less interpretable.

In addition, the limitation of this experiment is that the result of reflecting the two data in the production plan shows that the difference between operation time and operation quality has been affected, but it is unclear whether the efficiency of the production plan has improved, and only the change is known. In order for the results of this experiment to be meaningful, it is very important to involve the manufacturer's production managers in order to understand the results, even if we provide quantitative metrics of workload and time. The result of the production plan is a trade-off between all factors. Fluctuations in these values affect the overall production line balance and capacity. To overcome and adapt to this problem, it is important to involve the manufacturer's production management experts. To overcome the limitations of data forecasting in production planning, manufacturers can consider the following methods. Utilizing multiple data sources and applying machine learning and predictive models is the basis of this experiment. Establish a production plan based on the predicted data and simulate the production plan according to changes through scenario analysis so that you can make optimal decisions. It is also necessary to actively utilize the knowledge and experience of experts to overcome the limitations of data forecasting. While it is difficult to completely overcome the limitations of data forecasting, forecast accuracy can be improved by combining the above methods. Expert input can improve the reliability and relevance of forecast data by incorporating valuable domain-specific knowledge.

6. Conclusions

The focus is shifting from mass production to low-volume, high-variety production. Moreover, consumer needs are diversifying further. In addition, the recent COVID-19 pandemic has affected the supply chain and production planning, thereby requiring more complex demand and resource management. Therefore, it is necessary to conduct research on forecasting the fluctuating data to satisfy these changing conditions. Machine learning for predicting and reflecting the volatile baseline information during planning can be an indicator for effective manufacturing production planning. When the physical manufacturing environment changes, such as through manual processes, new development processes, new facilities, and new product development, the existing master data may become less reliable and need to be updated. Inaccurate data can result in sales losses and high production costs.

Based on the experiments in this study, we predicted two master data points using a machine-learning model. It was observed that when production is planned based on the predicted values, the work time and production volume could differ significantly from those in the plan formulated with the original master data. Experiments have revealed that the two datasets studied (yield and tact time) are important master datasets for production planning. Yield is a measure of the quantity of products manufactured to the desired quality standard in a production run. The tact time is a metric that indicates the frequency at which a product should be produced during the production process. The difference between operation quality and time is a factor that has an absolute impact on the utilization rate and output of the entire process, and through this experiment, it was confirmed that the predicted data show a difference in results and can be improved in terms of production management. If the difference can be applied as a reference for production management, it can be used to help predict and manage the master data.

The previously selected master data information predicts only two-attribute data and does not reflect time-series trends. This limits the scope of the data and prediction results. To overcome these limitations, research is needed to utilize the raw data that is generated at manufacturing sites but not being utilized. In the development of current technology, a lot of raw data is generated from sensors at manufacturing sites. Of course, this data is useful in many areas, but there is also a significant amount of data that is lost. There are studies that have applied machine-learning algorithms to monitor sensor data in order to improve acidity, reduce costs, and enhance safety [38].

As further research, we plan to test other algorithms, compare results, and apply AI models that improve performance or results. It is expected that by aggregating multiple data and managing a lot of master data using AI models, users will be able to utilize more pure data with less intervention. By learning from multiple data sources and analyzing time-series, it will be able to predict future data for the entire planning horizon, rather than just a cutoff point, and generate data that more accurately reflects trends. It is hoped that this deeper research will result in manufacturing process systems that can proactively respond to the ever-increasing complexity of the supply chain.

Author Contributions: Conceptualization, H.S. and J.J.; Methodology, H.S.; Software, H.S.; Formal analysis, H.S.; Investigation, H.S.; Resources, I.G. and J.J.; Data curation, H.S. and Y.K.; Writing—original draft, H.S.; Writing—review and editing, H.S., Y.K. and J.J.; Visualization, H.S. and J.R.; Supervision, J.J.; Project administration, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the SungKyunKwan University and the BK21 FOUR (Grad uate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the SungKyunKwan University and the BK21 FOUR (Graduate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF) This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2018-0-01417) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation).

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Sharma, R.; Shishodia, A.; Gunasekaran, A.; Min, H.; Munim, Z.H. The role of artificial intelligence in supply chain management: mapping the territory. *Int. J. Prod. Res.* **2022**, *60*, 7527–7550. [CrossRef]
2. Toorajipour, R.; Sohrabpour, V.; Nazarpour, A.; Oghazi, P.; Fischl, M. Artificial intelligence in supply chain management: A systematic literature review. *J. Bus. Res.* **2021**, *122*, 502–517. [CrossRef]
3. Janes, G.; Perinic, M.; Jurkovic, Z. Applying improved genetic algorithm for solving job shop scheduling problems. *Tech. Gaz.* **2017**, *24*, 1243–1247.
4. Thomas, E.H.; Galambos, N. What satisfies students? Mining student-opinion data with regression and decision tree analysis. *Res. High. Educ.* **2004**, *45*, 251–269. [CrossRef]
5. Arinez, J.F.; Chang, Q.; Gao, R.X.; Xu, C.; Zhang, J. Artificial intelligence in advanced manufacturing: Current status and future outlook. *J. Manuf. Sci. Eng.* **2020**, *142*, 110804. [CrossRef]
6. Helo, P.; Hao, Y. Artificial intelligence in operations management and supply chain management: An exploratory case study. *Prod. Plan. Control* **2022**, *33*, 1573–1590. [CrossRef]
7. Oluyisola, O.E.; Bhalla, S.; Sgarbossa, F.; Strandhagen, J.O. Designing and developing smart production planning and control systems in the industry 4.0 era: A methodology and case study. *J. Intell. Manuf.* **2022**, *33*, 311–332. [CrossRef]
8. Ren, L.; Zhang, Z.; Zhao, C.; Zhang, G. Cloud-based master data platform for smart manufacturing process. In Proceedings of the Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications: 9th EAI International Conference, CloudComp 2019, and 4th EAI International Conference, SmartGIFT 2019, Beijing, China, 4–5 December 2019; 21–22 December 2019; Springer: Berlin/Heidelberg, Germany, 2020; pp. 163–170.
9. Suri, R. *Quick Response Manufacturing: A Companywide Approach to Reducing Lead Times*; CRC Press: Boca Raton, FL, USA, 2020.
10. Javaid, M.; Haleem, A.; Singh, R.P.; Suman, R. Significance of Quality 4.0 towards comprehensive enhancement in manufacturing sector. *Sens. Int.* **2021**, *2*, 100109. [CrossRef]
11. Terwiesch, C.; Bohn, R.; Chea, K. International product transfer and production ramp-up: A case study from the data storage industry. *R&D Manag.* **2001**, *31*, 435–451.
12. Zhao, X.; Nie, X. Status Forecasting Based on the Baseline Information Using Logistic Regression. *Entropy* **2022**, *24*, 1481. [CrossRef]
13. Xu, K.; Li, Y.; Liu, C.; Liu, X.; Hao, X.; Gao, J.; Maropoulos, P.G. Advanced data collection and analysis in data-driven manufacturing process. *Chin. J. Mech. Eng.* **2020**, *33*, 43. [CrossRef]
14. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable ai: A review of machine learning interpretability methods. *Entropy* **2020**, *23*, 18. [CrossRef]
15. Mikalef, P.; Gupta, M. Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Inf. Manag.* **2021**, *58*, 103434. [CrossRef]
16. Wang, C.N.; Yang, F.C.; Vo, N.T.; Nguyen, V.T.T. Enhancing Lithium-Ion Battery Manufacturing Efficiency: A Comparative Analysis Using DEA Malmquist and Epsilon-Based Measures. *Batteries* **2023**, *9*, 317. [CrossRef]
17. Peng, F.; Wang, Y.; Xuan, H.; Nguyen, T.V. Efficient road traffic anti-collision warning system based on fuzzy nonlinear programming. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 456–461. [CrossRef]
18. Patange, A.; Jegadeeshwaran, R.; Dhobale, N. Milling cutter condition monitoring using machine learning approach. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *624*, 012030. [CrossRef]
19. Kumar, S.; Gopi, T.; Harikeerthana, N.; Gupta, M.K.; Gaur, V.; Krolczyk, G.M.; Wu, C. Machine learning techniques in additive manufacturing: A state of the art review on design, processes and production control. *J. Intell. Manuf.* **2023**, *34*, 21–55. [CrossRef]
20. Tao, F.; Qi, Q.; Liu, A.; Kusiak, A. Data-driven smart manufacturing. *J. Manuf. Syst.* **2018**, *48*, 157–169. [CrossRef]
21. Haug, A.; Stentoft Arlbjørn, J. Barriers to master data quality. *J. Enterp. Inf. Manag.* **2011**, *24*, 288–303. [CrossRef]

22. Law, C.C.; Chen, C.C.; Wu, B.J. Managing the full ERP life-cycle: Considerations of maintenance and support requirements and IT governance practice as integral elements of the formula for successful ERP adoption. *Comput. Ind.* **2010**, *61*, 297–308. [CrossRef]
23. Osterrieder, P.; Budde, L.; Friedli, T. The smart factory as a key construct of industry 4.0: A systematic literature review. *Int. J. Prod. Econ.* **2020**, *221*, 107476. [CrossRef]
24. Guo, K.; Yang, M.; Zhu, H. Application research of improved genetic algorithm based on machine learning in production scheduling. *Neural Comput. Appl.* **2020**, *32*, 1857–1868. [CrossRef]
25. Janeš, G.; Ištoković, D.; Jurković, Z.; Perinić, M. Application of Modified Steady-State Genetic Algorithm for Batch Sizing and Scheduling Problem with Limited Buffers. *Appl. Sci.* **2022**, *12*, 11512. [CrossRef]
26. Chen, M.; Sharma, A.; Bhola, J.; Nguyen, T.V.; Truong, C.V. Multi-agent task planning and resource apportionment in a smart grid. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 444–455. [CrossRef]
27. González Rodríguez, G.; Gonzalez-Cava, J.M.; Méndez Pérez, J.A. An intelligent decision support system for production planning based on machine learning. *J. Intell. Manuf.* **2020**, *31*, 1257–1273. [CrossRef]
28. Lakshmanan, V.; Robinson, S.; Munn, M. *Machine Learning Design Patterns*; O'Reilly Media: Sebastopol, CA, USA, 2020.
29. Ayodele, T.O. Types of machine learning algorithms. *New Adv. Mach. Learn.* **2010**, *3*, 19–48.
30. Kulkarni, E.G.; Kulkarni, R.B. Weka powerful tool in data mining. *Int. J. Comput. Appl.* **2016**, *975*, 8887.
31. Behnood, A.; Behnood, V.; Gharehveran, M.M.; Alyamac, K.E. Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. *Constr. Build. Mater.* **2017**, *142*, 199–207. [CrossRef]
32. Patange, A.D.; Jegadeeshwaran, R. A machine learning approach for vibration-based multipoint tool insert health prediction on vertical machining centre (VMC). *Measurement* **2021**, *173*, 108649. [CrossRef]
33. Pham, B.T.; Ly, H.B.; Al-Ansari, N.; Ho, L.S. A comparison of Gaussian process and M5P for prediction of soil permeability coefficient. *Sci. Program.* **2021**, *2021*, 3625289. [CrossRef]
34. Lin, L.; Wang, Q.; Sadek, A.W. A combined M5P tree and hazard-based duration model for predicting urban freeway traffic accident durations. *Accid. Anal. Prev.* **2016**, *91*, 114–126. [CrossRef]
35. Landwehr, N.; Hall, M.; Frank, E. Logistic model trees. *Mach. Learn.* **2005**, *59*, 161–205. [CrossRef]
36. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
37. Weisberg, S. *Applied Linear Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 528.
38. Patange, A.; Jegadeeshwaran, R.; Bajaj, N.; Khairnar, A.; Gavade, N. Application of machine learning for tool condition monitoring in turning. *Sound Vib.* **2022**, *56*, 127–145. [CrossRef]
39. Patange, A.D.; Pardeshi, S.S.; Jegadeeshwaran, R.; Zarkar, A.; Verma, K. Augmentation of Decision Tree Model Through Hyper-Parameters Tuning for Monitoring of Cutting Tool Faults Based on Vibration Signatures. *J. Vib. Eng. Technol.* **2022**, 1–19. [CrossRef]
40. Bueno, A.; Godinho Filho, M.; Frank, A.G. Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Comput. Ind. Eng.* **2020**, *149*, 106774. [CrossRef]
41. Nam, S.; Shen, H.; Ryu, C.; Shin, J.G. SCP-Matrix based shipyard APS design: Application to long-term production plan. *Int. J. Nav. Archit. Ocean. Eng.* **2018**, *10*, 741–761. [CrossRef]
42. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM Sigkdd Explor. Newsl.* **2009**, *11*, 10–18. [CrossRef]
43. Lantz, B. *Machine Learning with R: Expert Techniques for Predictive Modeling*; Packt Publishing Ltd.: Birmingham, UK, 2019.

applied sciences

MDPI

Article

Optimisation of Buffer Allocations in Manufacturing Systems: A Study on Intra and Outbound Logistics Systems Using Finite Queueing Networks

Mohamed Amjath [1], Laoucine Kerbache [1,2,*], James MacGregor Smith [3] and Adel Elomri [1]

1 Division of Engineering Management and Decision Sciences, College of Science and Engineering, Hamad Bin Khalifa University, Doha 34110, Qatar; mamjath@hbku.edu.qa (M.A.); aelomri@hbku.edu.qa (A.E.)
2 Information Systems and Operations Management, HEC Paris, 78351 Jouy-en-Josas, France
3 Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA 01003, USA; jmgsmith@acad.umass.edu
* Correspondence: lakerbache@hbku.edu.qa

Abstract: Optimal buffer allocations can significantly improve system throughput by managing variability and disruptions in manufacturing or service operations. Organisations can minimise waiting times and bottlenecks by strategically placing buffers along the flow path, leading to a smoother and more efficient production or service delivery process. Determining the optimal size of buffers poses a challenging dilemma, as it involves balancing the cost of buffer allocation, system throughput, and waiting times at each service station. This paper presents a framework that utilises finite queueing networks for performance analysis and optimisation of topologies, specifically focusing on buffer allocations. The proposed framework incorporates a finite closed queuing network to model the intra-logistics material transfer process and a finite open queueing network to model the outbound logistics process within a manufacturing setup. The generalised expansion method (GEM) is employed to calculate network performance measures of the system, considering the blocking phenomenon. Discrete event simulation (DES) models are constructed using simulation software, integrating optimisation configurations to determine optimal buffer allocations to maximise system throughput. The findings of this study have significant implications for decision-making processes and offer opportunities to enhance the efficiency of manufacturing systems. By leveraging the proposed framework, organisations can gain valuable insights into supply chain performance, identify potential bottlenecks, and optimise buffer allocations to achieve improved operational efficiency and overall system throughput.

Keywords: buffer allocations; finite queueing networks; GEM; simulation; optimisation; blocking

1. Introduction

The evolutionary trajectory of supply chain systems has given rise to the realisation that it is no longer a mere chain but a complex network [1]. Consequently, a novel term has been introduced to refer to conventional supply chains as demand networks [2]. The contemporary supply chain eco-system has undergone significant transformations over time, and supply chain systems are now widely regarded as sophisticated network systems. This evolution can be attributed to several factors, including the growth of trade and the economy, technological disruptions, increased product specialisation, and changing customer landscapes [3]. These changes have led to unprecedented transparency for all stakeholders, promoting excellence in the supply chain domain [4].

The supply chain industry has been transforming significantly due to increasing transparency across supply chains and networks. Despite this progress, organisations still require a more structured and systematic flow of information within and across their supply chains to better understand and have situational awareness of their systems [5].

Supply chain mapping is considered an ideal tool for visualising the network concerning different layers across and through the system [6]. The supply chain mapping is developed for several reasons, including performance measurement [7,8], re-configurations to achieve continuous improvements [9], integration with technological advances [10,11], and the ability to foresee supply chain risks and challenges posed by natural disasters, geopolitical instabilities, and pandemics [12]. According to MacCarthy et al. [6] supply chain mapping can be developed to clarify information at different levels, and Figure 1 illustrates a classification of hierarchies of supply chain mappings according to the relevant domains and focus areas.

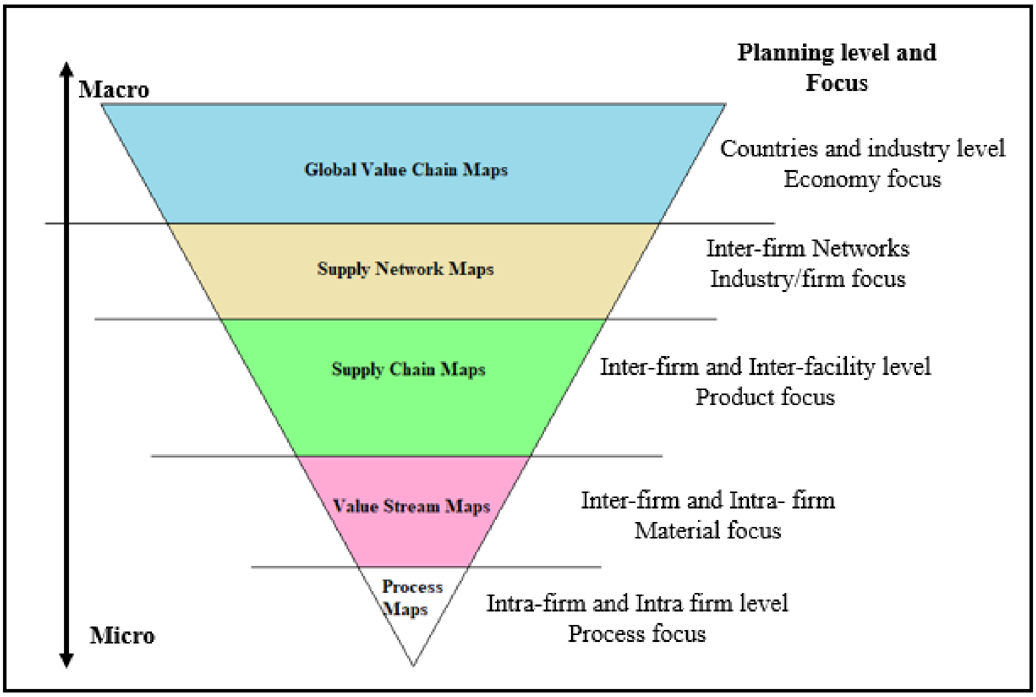


Figure 1. Classification of supply chain mapping hierarchy concerning planning levels and focus.

Global value chain maps serve as valuable tools for organisations to position themselves within the global structure and gain an understanding of the industries and markets in which they operate [13]. These maps provide a macro-level perspective and offer a high-level understanding of supply chain systems. A supply network map encompasses the supply chains that make up an industry, including the roles of regulatory bodies and other facilitating institutions, to gain an overall understanding of the industry's structure [14]. Supply chain maps are subsets of supply network maps that provide greater visibility into each element, such as inbound and outbound logistics, making the organisation the focal point of observation [15].

Moreover, supply chain maps are developed to illustrate the flow of products and services from the organisation's perspective. Value stream maps (VSM) provide information on the material, and information flows throughout the different stages of production, culminating in the final delivery of the product to the customer. VSMs have become a popular tool under lean manufacturing systems to identify and manage non-value-adding activities in the production processes [16–18]. Process maps are a conventional mapping tool used to understand the sequence of processes or tasks necessary to produce one unit of product or service [19]. Industry practitioners use time and motion studies and process maps interchangeably.

Value stream mapping is a widely used tool in the manufacturing and service industry to visualise and analyse the flow of materials and information and to comprehend the underlying business processes from an organisation-centric perspective. VSM is particularly effective in promoting cross-functional communication and collaboration within an organisation while providing opportunities for continuous improvement. Additionally, VSM is recognised for facilitating lean manufacturing practices, which help identify and eliminate waste and non-value-adding activities. The resulting efficiency improvements can help organisations track and optimise lead time, ultimately enhancing their overall performance. VSM provides the basis for developing supply chain maps, whereas it gets fed by the process mapping systems.

The interconnection between process topological design and VSMs lies in their ability to provide crucial insights into the structure and flow of activities within a system, particularly within the context of supply chains. Process topological design encompasses the layout, configuration, and connectivity of components or nodes in a system, aiming to attain desired performance objectives. Concurrently, VSMs offer a comprehensive understanding of the material and information flows within a specific process or value stream, aiding in identifying waste, inefficiencies, and improvement opportunities.

Buffer allocation, an integral aspect of topology design, plays a significant role in managing and optimising the flow of materials and information. In manufacturing systems and logistics networks, buffer allocation decisions are often made within the broader context of topology design. This entails determining the appropriate number, size, and placement of buffers at strategic locations throughout the system, aiming to ensure a balanced flow of goods or materials, minimise bottlenecks, and optimise system performance [20–22]. Poor buffer allocations in manufacturing and service settings can significantly affect operational performance. These consequences include reduced system throughput, as inadequate buffer allocations lead to congestion, bottlenecks, and delays in material transfers or service delivery. Longer lead times are another result, as insufficient buffer capacity increases waiting times and queuing delays at service stations [23]. Poor buffer allocations can also lead to higher inventory levels, as buffers are not appropriately sized or located, resulting in increased holding costs and potential obsolescence. Moreover, resource utilisation becomes imbalanced, with some areas being utilised and others underutilised, leading to inefficiencies and wasted resources [24]. Inefficient resource allocation is a consequence, with misalignment between resource capacity and demand patterns. Ultimately, better buffer allocations can result in customer satisfaction due to delays, disruptions, and unreliable performance. To mitigate these consequences, careful analysis and optimisation of buffer allocations, considering demand patterns, process variability, and resource capacities, are essential. Implementing a well-designed buffer allocation strategy can help minimise these negative consequences and improve overall operational performance [25].

By integrating VSMs into the process topological design, the current state of the process is visually represented, enabling the identification of critical paths, information flows, material flows, and interdependencies between different stages or workstations. This understanding of the process flow and interdependencies is then utilised to design an optimised process topological configuration. Buffer allocation decisions align with the identified bottlenecks, aiming to strategically allocate buffers or storage capacities to manage and optimise the flow of materials and information throughout the system.

Consequently, process topological design and value stream maps are interconnected, providing essential insights into the structure and flow of activities within a system. Buffer allocation, an integral component of topology design, facilitates efficient material and information flow management by strategically placing buffers at appropriate locations to balance the system, minimise bottlenecks, and optimise overall system performance [26].

This study aims to develop a process mapping approach that utilises finite queueing networks to analyse material logistics systems' performance within manufacturing facilities. Specifically, the focus is on the intra-logistics process, which involves transporting raw materials to different feeding points based on demand, and the outbound logistics process, which encompasses preparing finished goods for shipment to diverse customers. This research uses finite queueing network models to estimate key performance metrics such as waiting time, resource utilisation, and throughput. Finite queueing networks have proven valuable tools for modelling operations in manufacturing setups and determining appropriate buffer sizes. These networks capture the flow of entities (such as materials or customers) through a series of interconnected queues, representing the various stages or service stations within the manufacturing process. By incorporating factors such as arrival rates, service times, and queue capacities, finite queueing networks enable researchers and practitioners to simulate and analyse real-life scenarios more accurately [27,28].

Compared to infinite systems, which assume an infinite capacity for queues, finite queueing networks provide a more realistic representation of operational constraints and resource limitations. Due to physical and cost constraints, manufacturing setups typically have finite capacities for machines, workstations, and buffers. Researchers can model scenarios closely resembling manufacturing environments by considering these finite capacities [29].

By leveraging finite queueing networks, researchers can simulate different buffer allocation strategies, evaluate their impact on performance metrics such as throughput, waiting times, and resource utilisation, and ultimately determine the optimal buffer sizes for optimal operations. These models allow for experimentation and optimisation, enabling decision-makers to make informed choices regarding buffer allocations that balance operational efficiency, cost-effectiveness, and customer satisfaction [30,31].

This research seeks to make a scholarly contribution by delving into optimising buffer allocations in the material transfer process involving a homogeneous fleet of trucks. The study concentrates on two crucial material handling processes: Inter-facility transfer and outbound logistics. Notably, the significance of this study lies in its pioneering approach to buffer allocation problems within an organisational context, taking into account both ends of the logistics process. This research unfolds in two distinct phases, with the initial phase focusing on inter-facility material transfer and the second phase dedicated to outbound logistics. While prior studies have explored buffer allocation problems, they have predominantly concentrated on a singular logistics process. Consequently, this study takes a critical stride towards an integrated analysis of logistics processes within an organisational setup, specifically concerning buffer allocation problems.

The remaining sections of this study are organised as follows. Section 2 extensively reviews the existing literature about process mapping applications, buffer allocation problems, and the methodologies employed in previous studies. This section provides a comprehensive examination of prior research to establish the foundation for the current study. Section 3 explains the methodology adopted for the present investigation, outlining the key steps and procedures involved. Section 4 offers a meticulous analysis of the numerical experiments conducted using the proposed approach. This section incorporates a case study to demonstrate the method's effectiveness in practical scenarios. Section 5 concludes the study by summarising the key findings and contributions.

2. Literature Review

This section comprehensively examines the applications of finite queueing network models within process improvement, presenting significant insights gleaned from a sequence of empirical investigations conducted on real-world industrial operations. Moreover, studies related to buffer allocation problems and the employed methodologies, resulting outcomes, and significant ramifications are elucidated in detail. Moreover, particular emphasis is placed on underscoring the scholarly contributions of this study in advancing the optimisation of manufacturing processes, specifically pertaining to material logistics.

2.1. Process Maps and Applications

In a study conducted by Eleftheriadis and Myklebust [32], process maps were used to model the manufacturing process with process owner details to implement a zero-defect manufacturing environment. Similarly, Sharma [33] utilised process maps to analyse the welding process in a manufacturing facility and formulated a multi-criteria optimisation problem to make the process more sustainable. The optimisation problem was solved using meta-heuristics.

Araki et al. [34] and Mutua et al. [35] used process maps integrated with an optimisation problem to determine the fabrication parameters in a laser melting process. Bajaj et al. [36] and Wang et al. [37] utilised process maps in laser melting manufacturing processes to achieve reduced defect rates and identify the root causes of quality issues. Ponticelli et al. [38] presented a fuzzy genetic algorithm to optimise process control functions in a metal foam manufacturing facility. The process map was used to configure the operational parameters to improve the process and reduce defect rates.

He et al. [39] utilised process maps in a polymer dispensing system to determine the process planning parameters. Sabzi et al. [40] used process maps for alloy design and process optimisation. The authors presented a framework to design alloy production and determine the optimal solidification and thermal formation and then used a genetic algorithm to solve the optimisation problem. In a study conducted by Mejri et al. [41], process mapping was used to visualise a detergent manufacturing process. The authors used process re-engineering and innovation to optimise the production process while eliminating uncertainty.

In the additive manufacturing field, process maps are widely used to improve current processes and enhance efficiency, as demonstrated in studies conducted by Gui et al. [42], Akbari et al. [43], and Patel et al. [44]. Kumbhar et al. [19] used process maps integrated with digital twin models to detect bottlenecks in manufacturing systems for the diagnosis and improvement of the system.

2.2. Order Releasing and Scheduling Using Heuristic Based on Drum Buffer (DRB) Rope Technique

Bisogno et al. [45] explored the application of the theory of constraints (TOC), specifically the DBR technique, to improving process performance in healthcare services. Customising DBR for scheduled patient flows and introducing DBR for unscheduled patient flows, the study provided rules to control patient throughput, considering the trade-off between minimising flow times and maximising throughput volumes. Simulation experiments demonstrated the impact of TOC DBR rules on this trade-off.

Thürer and Stevenson [46] examined bottleneck shiftiness and order release methods, revealing that downstream bottleneck shifts had adverse effects on performance while upstream shifts had minimal impact. The distance between the actual and assumed bottlenecks had negligible performance implications, offering valuable insights for DBR and similar release methods. Thürer and Stevenson [47] further explored different sequencing and dispatching rules to enhance the performance of the DBR scheduling mechanism. The researchers found that prioritising the shortest bottleneck processing time during high load periods significantly improved performance.

Researchers are now exploring flow shop configurations, focusing on bottleneck management and connections to the theory of constraints (TOC). The drum buffer rope (DBR) methodology has been developed to address these challenges as a production planning and control tool within this context [48]. Yue et al. [49] presented a heuristic approach based on the DBR method to address the challenges of order releasing and multi-item scheduling in MTO production systems. The proposed approach aims to enhance productivity and efficiency in MTO companies by considering dynamic customer demands and capacity-constrained resources. Through experimentation on different problem types based on due date tightness and product demand, the performance of the proposed heuristic is compared with other well-known heuristic methods from the literature. The results indicate that the proposed DBR-based heuristic outperforms the competitors, particularly when an optimal buffer size is adopted, offering significant improvements in order release and scheduling effectiveness.

2.3. Multi-Level Rolling Horizon Planning and Scheduling Using DBR Approach Considering Material Constraints

The advent of Industry 4.0 concepts has led to a growing trend of digitization in industries. Consequently, traditional planning and scheduling approaches need to be revised to address the challenges posed by this new digital landscape.

Saif et al. [50] proposed a drum buffer rope-based heuristic algorithm (DBR-HA) for efficient planning and scheduling in mixed-model production industries aiming to implement Industry 4.0. The algorithm considered shifting bottleneck resources and maximised the use of capacity constraint resources, providing effective plans and schedules for each planning horizon. Lin et al. [51] integrated TOC and ERP in production planning to improve quality and cost efficiency. Utilising the DBR method, the study synchronised production, identified bottlenecks, allocated resources effectively, and enhanced production performance. A practical case demonstrated the successful integration and implementation of TOC and ERP in production planning.

Prasetyaningsih et al. [52] addressed an imbalance problem in a shoe company's production lines by applying the TOC approach, utilising linear programming for production planning optimisation, and DBR for flow control. By considering the output gap between production lines, buffers were determined, and three solution alternatives were proposed, resulting in a substantial reduction of the imbalance issue at the shoe production lines. Telles et al. [53] examined the effects of implementing DBR on the efficiency of three engineering-to-order (ETO) production lines in an aerospace manufacturing context. The analysis, conducted longitudinally through a case study, utilised data envelopment analysis (DEA), the Wilcoxon test, and analysis of variance (ANOVA) to assess the impact of DBR. The findings revealed a significant efficiency increase of up to 19% following the implementation of DBR. Liu et al. [54] incorporated the theory of constraints and leveraged the DBR mechanism to develop multi-level planning and scheduling strategies in the context of mixed-model production. Additionally, the researchers propose a multi-level planning heuristic (MLPH) that employs DBR and priority rules to achieve efficient planning and scheduling. The approach considers material requirement planning and optimally utilises capacity-constrained resources (CCR) to the fullest extent.

2.4. Buffer Allocation Problems

Cruz et al. [55] presented an original methodology to address the buffer allocation and throughput trade-off problem in finite queueing networks. A specialised multi-objective genetic algorithm approximates the Pareto optimal set of solutions. Computational results demonstrate the efficiency and efficacy of the proposed methodology, highlighting the impact of the coefficient of service time variation on buffer allocation decisions and the dependence of buffer allocation on target throughput. Smith et al. [56] addressed optimising topological network design for multi-server queueing networks. Series, merge, and split topologies are examined using approximation and iterative search methods to estimate performance and determine optimal buffer allocation. The impact of the coefficient of variation on buffer allocation is highlighted, and computational results illustrate emerging buffer patterns within different topologies.

G. Wang et al. [57] compared three layout structures in automotive engine shops to maximise profit based on throughput and buffer investment costs. The analysis utilises queueing network models with finite buffers and unreliable machines, considering heterogeneous service times and exponentially distributed failure and repair times. The study employed approximation methods and a gradient search approach to determine the optimal buffer allocation. Zhang et al. [58] modelled mould manufacturing using OQNs with finite buffers. Based on queuing theory approximations, the proposed methods effectively evaluate system performance and demonstrate feasibility for large-scale practical problems. The findings provide valuable insights for system design, resource planning, buffer allocation, and capacity configuration. MacGregor Smith [59] presented an approach to determine the buffer allocation of a finite system using a queueing network decomposition methodology combined with a nonlinear sequential quadratic programming algorithm. The joint optimisation problem maximises throughput by treating the population as a constraint. Li et al. [60] developed analytical formulas to estimate the throughput of a reliable production line with exponential service times and finite buffers. The formulas apply to approximately balanced lines with similar processing times and provide upper bounds for throughput.

Cruz et al. [61] presented a study of optimising performance in finite single-server acyclic queueing networks using a multi-objective methodology to minimise buffer areas and service rates while maximising throughput. The proposed approach utilises a simulated annealing algorithm to redistribute buffer spaces and improve throughput without compromising overall capacity. Pedrielli et al. [62] introduced the discrete event optimisation (DEO) methodology for the simultaneous simulation and optimisation of manufacturing systems for buffer allocation in a production line. DEO utilises mathematical programming to approximate mixed integer linear programming models, providing a formal approach for simulation optimisation in queueing systems.

Yu et al. [63] addressed the optimisation of buffer allocation in AMHS in intelligent manufacturing workshops, highlighting the need for more integration between production and MHS in existing research. The study proposes a flow shop model with capacitated batch transports, utilising an OQN with blocking. An approximation method is presented to compute performance measures, and an iterative optimisation algorithm is developed for determining optimal buffer allocation. Hu et al. [64] presented a study identifying the optimal task assignment policy for maximising long-term average throughput. It provides insights into the preferred server coordination policies based on buffer allocation and numerical comparisons and generalisations for longer queueing lines.

A co-occurrence network (Figure 2) in the relevant literature provides insights into their interconnectedness and identifies important research trends and relationships. For example, the co-occurrence of keywords such as manufacturing, material handling, and machine shop practice suggests a strong association between these concepts in buffer allocation and queueing networks. Performance evaluation and throughput are often explored together, indicating the significance of evaluating system performance in terms of throughput. The co-occurrence of keywords such as system analysis, topology, and blocking implies the importance of studying the system's structure and characteristics, including network topology and the impact of blocking on performance. Moreover, keywords such as integer programming and genetic algorithms suggest utilising optimisation techniques for solving buffer allocation problems in queueing networks. These methods are commonly employed to find optimal solutions by balancing conflicting objectives.

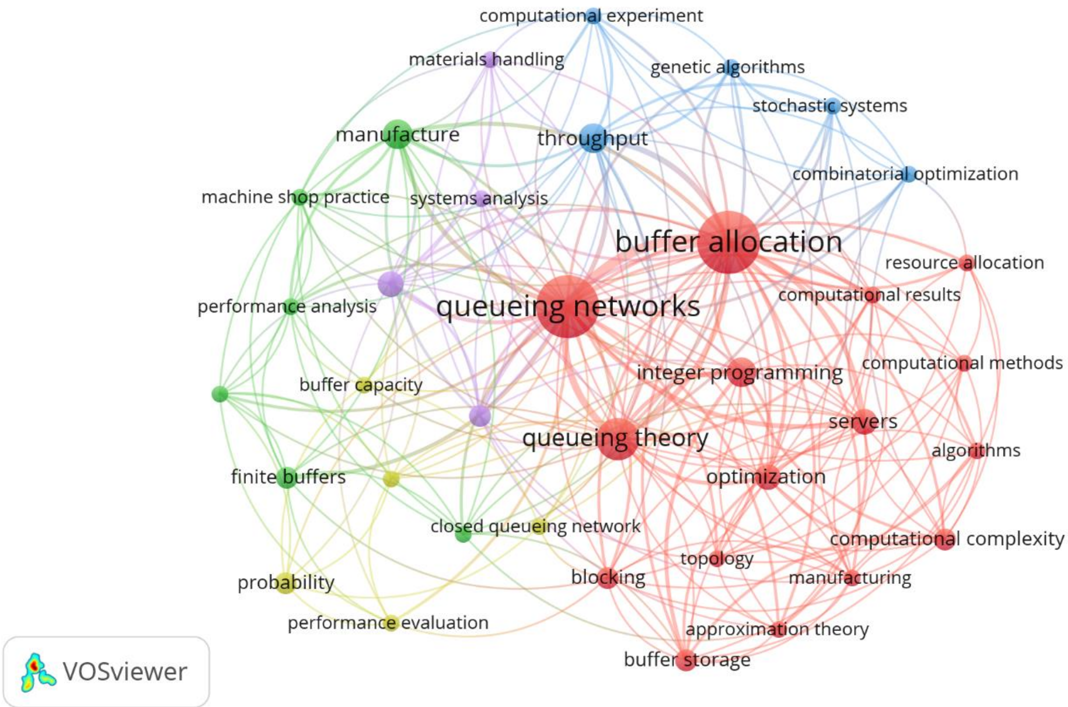


Figure 2. Co-occurrence network of keywords in buffer allocation and queueing networks for manufacturing systems.

Uniform and Non-Uniform Buffers

Uniform or homogeneous buffers possess equivalent capacities or sizes throughout the queueing network. Conversely, non-uniform buffers, also called heterogeneous buffers, exhibit varied capacities or sizes across different buffers in the queueing network, enabling each buffer to accommodate different entities. From an optimisation standpoint, the suitability of uniform buffers in a queueing network relies on the specific goals and requirements of the analysed system [65]. While uniform buffers can simplify the analysis, they may only sometimes represent the optimal choice regarding performance or resource utilisation. Uniform buffers foster balanced resource utilisation across the network, which can be advantageous in scenarios prioritising uniformity to ensure fairness and prevent bottlenecks. However, in certain situations, non-uniform buffers may offer superior resource utilisation. For instance, when demand or arrival rates significantly differ across different network parts, allocating larger buffers to high-demand areas can avert congestion and optimise resource allocation. Uniform buffers may not be optimal for minimising queueing delays. In the presence of heterogeneous arrival rates or service times, non-uniform buffers facilitate the allocation of buffer capacity based on demand at each network stage. By allowing larger buffers at bottleneck points or areas with high variability in arrival rates, non-uniform buffers can diminish queueing delays and enhance the overall system's performance [66]. The cost associated with buffer size plays a vital role in optimisation. Uniform buffers simplify resource allocation and buffer management but can result in over-provisioning in specific network segments.

Conversely, appropriately sizing non-uniform buffers based on demand patterns can optimise resource utilisation and potentially reduce costs by eliminating unnecessary buffer capacity. Non-uniform buffers offer greater flexibility and adaptability to dynamic changes within the system. In situations where the demand pattern or service times vary, non-uniform buffers enable better adjustment to changing requirements. This flexibility leads to improved performance and responsiveness compared to uniform buffers, which may need to be more adaptable to varying conditions [67].

2.5. Solving Finite Queueing Networks

Solving finite queueing networks presents more significant challenges than solving infinite networks due to several key factors. Firstly, finite networks have a complex and expansive state space that grows exponentially with the number of nodes, buffers, and entities. This complexity hinders direct analysis and solutions using traditional methods [68]. Secondly, interactions and dependencies among nodes in finite networks introduce intricate dynamics, making it more challenging to understand entity flow, bottlenecks, and the influence of routing strategies compared to the simplified assumptions in infinite networks. Additionally, evaluating performance measures such as queue lengths, waiting times, and resource utilisation becomes more complex due to interdependencies among nodes and the variability of system parameters [69]. Non-exponential service time distributions in practical scenarios further complicate the analysis, as traditional methods assume exponential distributions [70]. Adapted analytical techniques or alternative methodologies are required to address this challenge. Finally, as the size and complexity of finite queueing networks increase, analytical solutions may become intractable, necessitating numerical methods [71], approximations [72–75], or simulations [20,76] that demand significant computational resources and time. In light of these complexities, researchers employ a range of methodologies and techniques, such as approximation methods, decomposition techniques, numerical methods, and simulations, to overcome the challenges and gain meaningful insights from finite queueing networks.

Blocking Phenomenon

According to Smith [77], three types of blocking can occur in a two-stage queueing network: blocking after service (BAS), blocking before service (BBS), and repetitive service blocking (RSB). In the case of BAS, a customer or entity remains on the server until they are serviced, and they can only proceed to the downstream block if there is an empty space (buffer) available. BAS ensures that customers are not prematurely removed from the system, allowing them to complete their service before moving forward in the network. In the case of BBS, customers or entities are prevented from entering the server or queue before receiving service due to factors such as limited buffer space or predetermined constraints. This blocking occurs at the server's entrance, resulting in a loss of service opportunity. On the other hand, RSB refers to a situation where a customer repeatedly receives service from a server, preventing other waiting customers from being served. RSB typically arises when a customer requires multiple service instances consecutively, effectively monopolising the server's attention and delaying service for other waiting entities.

2.6. Scholarly Contributions and Future Extensions of the Study

This study aims to make a scholarly contribution by focusing on optimising buffer allocations in the material transfer process involving a homogeneous fleet of trucks. Specifically, the study addresses the optimal buffer allocation in two material handling processes: The inter-facility transfer and outbound logistics processes. These material handling processes were modelled using finite queueing networks. The networks are designed to accommodate constrained and unconstrained populations with generalised service times. By utilising the GEM approximation approach, the performance measures of the processes are evaluated, including cycle time, resource utilisation rates, and throughput levels. Given that the optimal buffer allocation problem is NP-hard, this study adopts a simulation optimisation approach to solve it. This approach considers the complex interactions and dependencies between various factors involved in the buffer allocation problem. In future extensions of this research, a comprehensive analysis of all logistics processes will be conducted to gain a holistic understanding of the supply chain across all stages. This integrated approach will provide valuable insights for optimising the overall performance and efficiency of the supply chain.

3. Methodology

This section elucidates the methodology implemented in the present study, focusing on modelling inter-facility and outbound logistics processes as finite queueing networks. The GEM approximation technique is elaborated upon, enabling the estimation of network performance measures. Additionally, a mathematical model is formulated to optimise buffer allocation within the network, followed by an explanation of the solution approaches employed to address the optimisation problem.

3.1. Modelling the Material Transfer Processes Using a Finite Queueing Network

The study adopts a rigorous technical approach to modelling material transfer operations involving a homogeneous fleet of trucks. The trucks carrying raw materials from storage to feeding points are represented as network jobs. Each sub-process within the material transfer operations is depicted as a node in the network. The inter-facility material transfer process encompasses a restricted number of assigned trucks, while the outbound logistics process allows for an unrestricted number of customer trucks. Therefore, the study uses finite CQN and finite OQN for the abovementioned cases. Different raw materials or product types are represented through multi-class jobs.

The stochastic nature of the service time at each node captured within the queues is modelled as M/G/1 queues, assuming a generally distributed service time. The queue discipline follows a first-come, first-serve policy. All stations within the network consist of a single-server configuration, and the movement between stations is modelled with an infinite number of servers.

The presence of limited buffer capacity at each node gives rise to the potential occurrence of blocking. In this study, the blocking after service (BAS) strategy is employed, wherein a job remains in the current node even after completion of service if it cannot be transferred to the succeeding node due to insufficient available space. Consequently, exponential service time distributions are unsuitable for this analysis, as they assume memoryless service durations following a constant exponential distribution. Within an exponential setting, jobs serviced at a given node promptly depart to the next node, assuming adequate space is available.

3.2. Generalized Expansion Method (GEM)

The GEM has emerged as a powerful approach for solving finite queueing networks and has demonstrated several advantages compared to other methodologies. GEM offers accurate performance analysis by considering finite capacities, blocking phenomena, and service time distributions [78]. It provides a robust framework for analysing complex systems with multiple interconnected queues, enabling researchers to derive analytical expressions for performance measures. Regarding implementation cost, GEM offers an advantageous position as it relies on existing mathematical techniques and equations without requiring extensive computational resources or specialised software [79]. Additionally, GEM is relatively easy to use, providing researchers with clear analytical insights into system behaviour and facilitating sensitivity analysis. While other methodologies, such as simulation-based approaches, may offer more flexibility in capturing system complexities and stochastic behaviour, they often require significant computational resources and expertise in simulation software [80]. Overall, GEM is a valuable methodology for accurate performance analysis of finite queueing networks, offering a cost-effective and user-friendly alternative to other methodologies.

The GEM, as proposed by Kerbache and Smith [73], comprises three fundamental steps: network reconfiguration, parameter estimation, and feedback elimination (Figure 3). This study employs GEM as an approximation technique to analyse the finite queueing network and the blocking phenomenon.

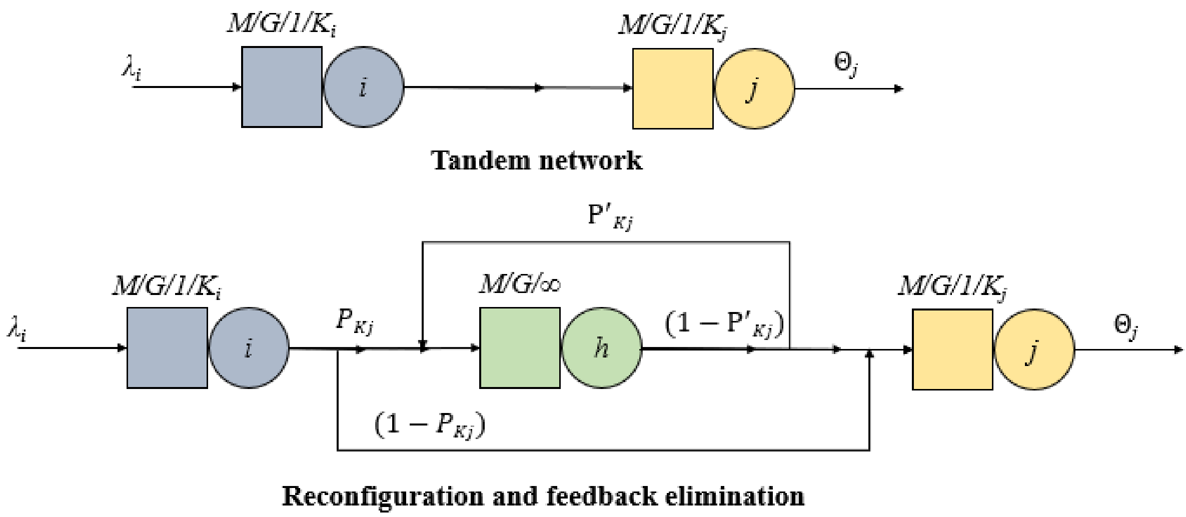


Figure 3. GEM steps for a tandem network (adapted from Kerbache and Smith [73]).

The first step, network reconfiguration, involves restructuring the original queueing network by introducing auxiliary nodes (holding nodes) and links to account for the blocking effects. This reconfiguration enables the representation of blocking occurrences within the network model. The second step, parameter estimation, aims to estimate the unknown parameters of the modified queueing network. This estimation step involves determining the routing probabilities, blocking probabilities, squared coefficient values, service times, and other relevant parameters crucial for accurately modelling the system behaviour.

The final step, feedback elimination, focuses on eliminating the feedback loops that may arise due to the reconfiguration. Feedback loops can lead to complexity in the analysis and may hinder the accurate estimation of performance measures. Removing these feedback loops can simplify and analyse the modified queueing network more efficiently. The following mathematical notations were employed to derive equations for the performance measures of the finite queueing network using GEM.

Notations and Definitions

M Number of nodes
Λ_j Input rate node i (i = 1,2,... M)
T Number of product classes t (t = 1,2,... T)
B_i Buffer capacity at node i excluding those in service
K_i Buffer capacity at node i including those in service
α_j The probability of jobs leaving the system from node j (j = 1,2,..., M)
C_j Cost of a buffer space in node j
L_{sj} Mean number of jobs (queue length) at queue s in node j
λ_h Arrival rate to holding node h
μ_h Mean service rate at holding node h
P_{Kj} Blocking probability that node j is at capacity K_i
P'_{Kj} Feedback blocking probability
Θ Mean throughput of the system
Θ_j Output rate at node j
U_j Utilisation rate at node j
S^2 Squared Coefficient of variation of the service time
$\rho = \frac{\lambda}{\mu}$ Server busy time as a proportion of arrival and service rate
$\rho_h = \frac{\lambda_h}{\mu_h}$ Server busy time as a proportion of arrival and service rate at node h
β Maximum budget allocated for buffer space allocation
D_s Mean waiting time at queue s

N Network population (for CQN)

An M/G/1 queue is a queueing system where customer arrivals follow a Poisson process (M), service times have a general distribution (G), and there is a single server (1) serving customers one at a time.

Based on the two-moment approximation technique, a closed-form approach can be used to calculate the blocking probability using the following equation [73,81].

$$P_{Kj} = \frac{\rho^{\frac{(2+\sqrt{\rho s^2}-\sqrt{\rho}+2(K_j-1)}{(2+\sqrt{\rho S^2}-\sqrt{\rho})}}(\rho-1)}{\rho^{\frac{(2+\sqrt{\rho s^2}-\sqrt{\rho}+2(K_j-1)}{(2+\sqrt{\rho S^2}-\sqrt{\rho})}}-1} \tag{1}$$

According to Zhang et al. [82], the output rate of node *j* can be calculated using the following for an OQN system.

$$\Theta_j = \Lambda_j\left(1-P_{Kj}\right) + \lambda_h\left(1-P'_{Kj}\right)^{\rho_h}\cdot\left(1-P_{Kj}\right)^{\rho_j} \tag{2}$$

According to Kerbache and Smith [26], the total throughput of node *j* in a multi-class finite CQN with total population of *N*.

$$\Theta_j = \frac{\sum_{i=1}^{M}\sum_{t=1}^{T}\left(\lambda_{ij}\cdot\left(1-P_{Kj}\right)\right)}{N} \tag{3}$$

According to Kerbache and Smith and Kerbachea and Smith [72,73] the utilisation rate at node *j* in OQN and CQN can be calculated using the following equations, respectively.

$$U_j = \frac{\lambda_j}{\left(\Theta_j\cdot K_j\right)} \tag{4}$$

$$U_j = \frac{\Theta_j}{\mu_j} \tag{5}$$

3.3. Formulation of Optimisation Problem

Buffer allocation in queueing networks offers the potential to optimise multiple objectives by considering various performance measures such as throughput, waiting time, resource utilisation, and system stability. However, achieving a harmonious equilibrium among these objectives poses challenges due to their inherent trade-offs.

The complexity of the buffer allocation problem arises from the need to determine the optimal allocation of finite buffer resources across different nodes within the network. The task involves striking a delicate balance in buffer sizes to maximise overall system performance while simultaneously addressing multiple objectives. For instance, increasing buffer sizes can alleviate congestion and enhance throughput, but it may also lead to prolonged waiting times. Conversely, reducing buffer sizes may enhance responsiveness but elevate blocking probabilities and diminish overall throughput.

In order to address the buffer allocation problem in a finite queueing network, an optimisation framework is formulated with a carefully designed objective function and constraints [83].

$$Maximise\ \Theta\ (K,\mu) \tag{6}$$

$$Maximise\ \Theta\ (N,K) \tag{7}$$

Constraints

$$\sum_{i=1}^{M} K_i\cdot C_i \le \beta \tag{8}$$

$$L_i \le K_i\ \forall\ i = 1,2,..M \tag{9}$$

$$1 \leq K_i \in \mathbb{Z} + \forall i = 1, 2, ..M \tag{10}$$

Equation (6) presents the system throughput as a function of arrival rates and buffer sizes in the OQN context. Similarly, Equation (7) demonstrates the throughput as a function of network population and buffer sizes in CQN. In this study, we will maintain constant arrival rates for OQN and network population for CQN to determine the optimal buffer allocation for each node, aiming to maximise throughput. However, we will conduct various scenarios in both cases, considering different arrival rates, network populations, and compositions to assess their impacts on the system's overall throughput. It is important to note that the objective function in both cases is subject to constraints. For instance, Equation (8) establishes an upper bound on the total number of buffers allocated due to budget limitations. Additionally, in a steady state, the average queue length should not exceed the buffer sizes of a given node (Equation (9)), and the number of buffers allocated must always be a positive integer with a minimum value of one (Equation (10)).

The optimisation problem of buffer allocation, known for its NP-Hard nature, poses a significant challenge in finding an optimal solution in a reasonable computation time [26,84]. Over the years, various solution approaches have been developed to tackle this intricate problem. Traditional methods, such as mathematical modelling and optimisation algorithms, have been widely employed to derive optimal buffer allocation strategies. However, due to the complexity and combinatorial nature of the problem, these approaches often require more computation and may only sometimes yield optimal solutions. As a result, simulation optimisation approaches have gained prominence in recent years. By combining the power of simulation and optimisation techniques, these approaches provide robust solutions with high accuracy, making them increasingly popular for addressing the buffer allocation optimisation problem.

The buffer allocation problem in this study was optimised using the built-in optimisation engine of AnyLogic software. The authors utilised a personal computer with 8GB of RAM and a core i3 7th Gen processor to execute the simulation experiments. The maximum iteration count was set to 5000 runs to ensure convergence towards optimal results. Figure 4 comprehensively illustrates the sequential steps in this study's AnyLogic optimisation solution approach.

The OptQuest optimisation engine in AnyLogic simulation software solves larger combinatorial problems by efficiently navigating search trees and identifying fruitful directions towards optimal solutions. It utilises advanced techniques such as genetic algorithms, simulated annealing, and tabu search to explore the solution space and prune unproductive paths [85].

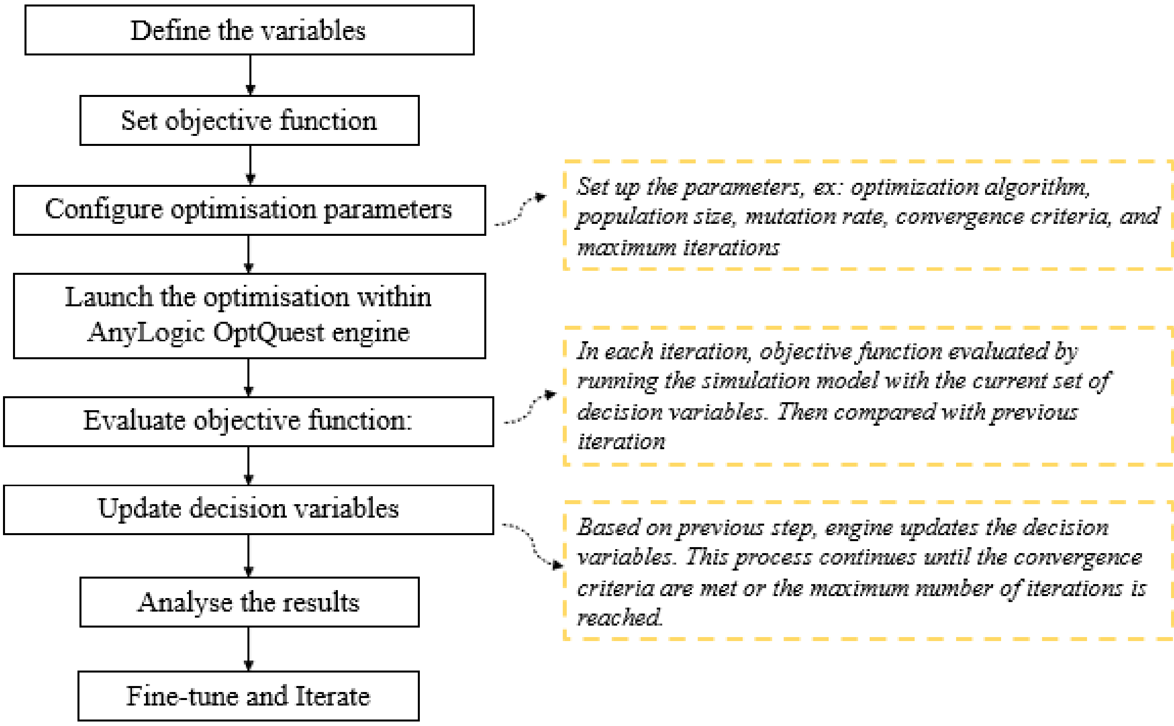


Figure 4. Steps of optimisation approach used in Anylogic software.

4. Case Study—Numerical Experiments

To demonstrate buffer allocation problems, this section presents a case study that utilizes numerical experiments to investigate and evaluate the proposed methodology in inter-facility material transfer operations and outbound logistics processes. These experiments aim to validate the effectiveness and efficiency of the methodology in optimising the buffer allocation problem within the intra-organisational material flows and the corresponding outbound logistics activities. We conducted the case study using a manufacturing facility with multiple workstations as the testbed. Applying the analytical method based on GEM, we formulated a finite queueing network model that considered various parameters, such as arrival rates, service times, and buffer capacities. By solving the GEM equations, we obtained analytical expressions for performance measures. Simultaneously, we developed a simulation model to replicate the dynamics of the facility and conducted multiple numerical experiments to collect data on system performance metrics. The results from the analytical method and simulation were compared to assess the effectiveness of the proposed methodology in optimising buffer allocation. This empirical illustration demonstrates the practical application of the proposed methodology. It showcases its efficacy in addressing the buffer allocation problem in the context of intra-organisational material flows and outbound logistics processes.

4.1. Inter-Facility Material Transfer Process

SM is a specialised steel manufacturing firm that produces steel rebars for the domestic market. To manufacture steel billets, which serve as the primary inputs for producing steel rebars, SM employs an electric arc furnace (EAF) system. The billet production process involves utilising various materials, including scrap materials, hot briquette iron (HBI), and specific additives such as carbon, chromium, and deoxidisers such as aluminium or silicon. Daily, SM acquires scrap materials from the domestic market, which are stored in different forms and purities in the SM storage yard. These forms include heavy metal scrap and shredded scrap. Additionally, SM incorporates HBI and direct reduced iron (DRI) as raw materials in the billet production process. The selection of billet grades is determined by evaluating the quality and percentage of the scrap materials. To simplify our discussions in subsequent sections, we will utilise the notations A, B, and C to represent heavy metal scrap, shredded scrap, and HBI/DRI materials.

In the context of SM, the production of steel billets takes place on a daily basis, using materials A, B, and C. These materials are transported each day by a fleet of homogeneous trucks to the facility situated within the billet plant. Figure 5 illustrates the inter-facility material transfer operations undertaken to meet the demand requirements of the billet plant. Figure 6 presents the layout of the storage area and the billet plant, visually representing their spatial arrangement.

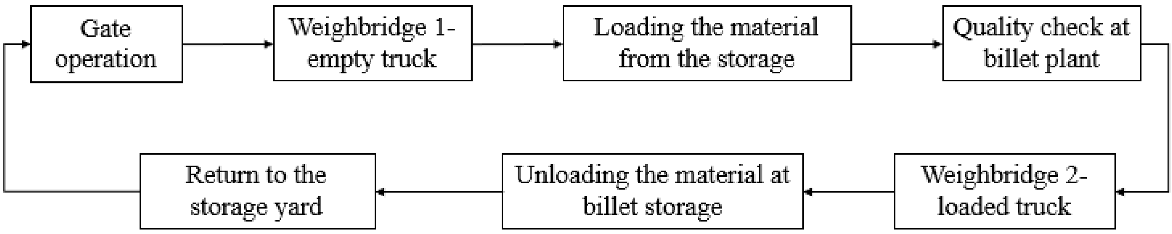


Figure 5. Process flowchart for the inter-facility material transfer operations.

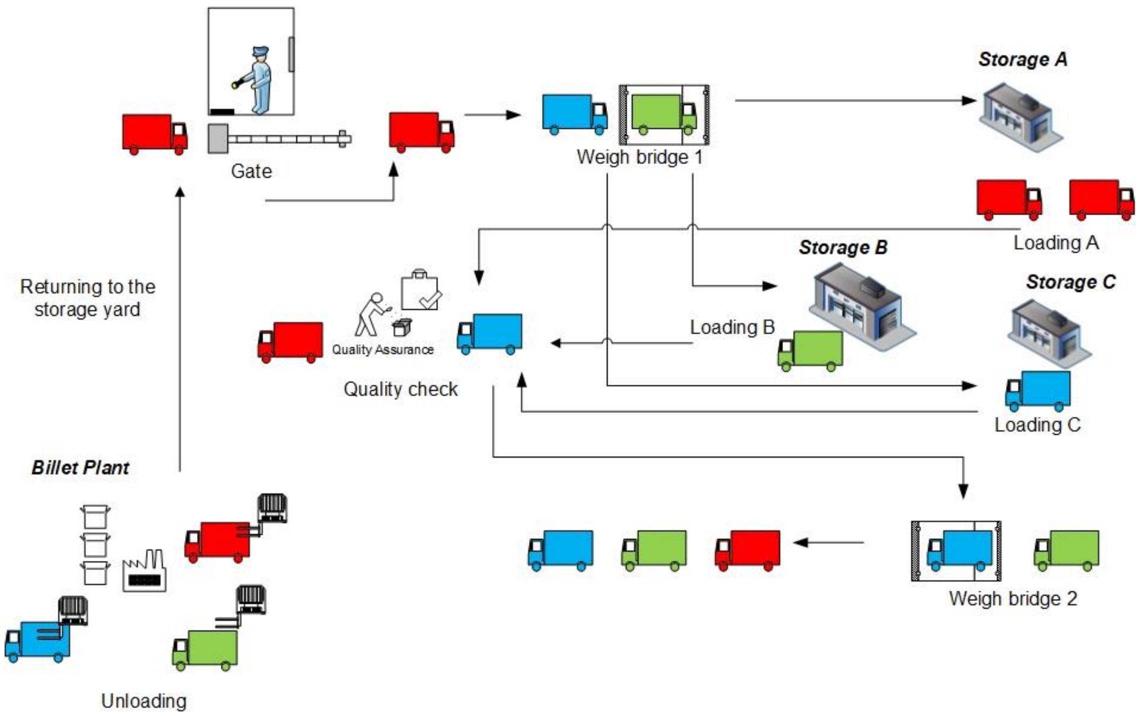


Figure 6. Layout plan of SM's raw material storages and billet plant.

As depicted in Figure 6, except for the loading and unloading service stations, all other stations are shared among the three types of material trucks. Truck servicing follows a first-come, first-served basis. The node details and average service time for each service station are presented in Table 1, denoted in minutes. Additionally, Table 2 provides the weight of a full truckload of each material.

Table 1. Node details and average service times.

Node #	Process	Aver. Service Time (min)	Node #	Process	Aver. Service Time (min)
1	Gate operation	1.5	7	Weigh bridge 2	1.5
2	Weigh bridge1	1.5	8	Unloading—A	2.5
3	Loading—A	6	9	Unloading—B	2
4	Loading—B	4	10	Unloading—C	3.5
5	Loading—C	7	11	Return trip	6
6	Quality check	2			

Table 2. Full truckload weights.

Raw Material (Product Class)	Full Truck Load (Tons)
A	7
B	3
C	9

Each service station within the system is managed by a single server, implying that each station has the capacity to serve only one truck at any given time.

In order to evaluate and optimise inter-facility material transfer operations, this study conducted performance measurements and buffer allocation optimisation exercises. The experiments were designed based on scenarios identified in Table 3. Additionally, various parameters were considered, including the cost of buffer space (C_i) and the maximum allotted budget for buffer space allocation (β), with values of EUR 200 and EUR 5000, respectively. All experiments were conducted to simulate a 24 h time-period operation.

Table 3. Scenarios used for optimisation and performance analysis experiments.

Scenario #	Number of Trucks			Total Number of Trucks
	A	B	C	
Scenario 1	3	3	3	9
Scenario 2	4	4	4	12
Scenario 3	6	6	6	18
Scenario 4	4	5	6	15
Scenario 5	6	4	5	15

4.1.1. Development of the DES Model for Inter-Facility Material Transfer Operations

The DES model for the inter-facility material transfer operations was developed based on the abovementioned details, as illustrated in Figure 7. Anylogic University edition (version 8.8.1) simulation software was employed. The model represents all service stations (nodes) as finite queue capacity service blocks. The decision variables, namely the buffer sizes for each node, were introduced as parameters in the optimisation configuration. The objective function was formulated to calculate the total throughput of the system. The trucks carrying three different types of materials were incorporated into the system through three sources. Each service block is coupled with a single resource pool to ensure the single-server phenomenon. Furthermore, the model accounted for the blocking phenomenon known as blocking after service (BAS), which occurs when a truck will not leave the station even after servicing if there is no space in the next node.

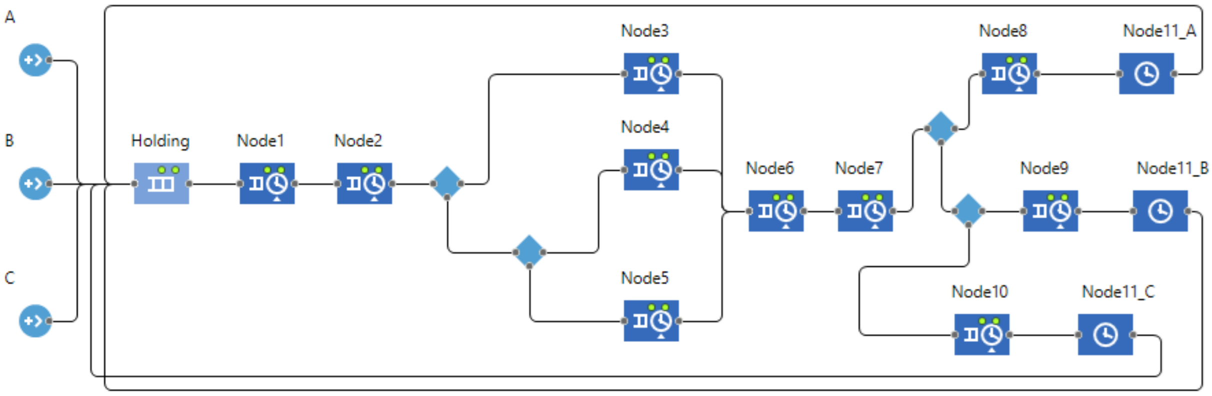


Figure 7. DES model for SM inter-facility material transfer operations.

4.1.2. Optimal Buffer Allocation

Table 4 presents the optimal buffer allocations obtained for each scenario investigated in the study. The allocations differ across scenarios, indicating the significance of tailoring buffer allocation strategies to specific operational conditions.

Table 4. Optimal buffer allocations in each scenario.

Scenario #	Optimal Buffer Allocation	Throughput (Tons of Materials Transferred)	Total # of Buffers Used
Scenario 1	4,1,4,3,1,3,2,2,1,1	3101	22
Scenario 2	1,1,3,2,2,4,3,2,2,1	3656	21
Scenario 3	1,3,2,4,1,1,2,7,2,1	3677	24
Scenario 4	1,1,1,1,1,10,1,1,1,1	3628	18
Scenario 5	2,1,6,2,1,5,1,1,3,3	3794	25

A uniform truck allocation was employed for all material types in the initial three scenarios. Furthermore, an analysis of the results reveals that an increase in the number of trucks allocated to each material type and the total number of trucks corresponded to an increase in throughput, quantified as the total tons of materials transferred. However, it is worth noting that this increase in throughput was accompanied by a potential decrease in the average throughput per truck. In scenarios 4 and 5, different truck allocations were implemented for each material type while maintaining a constant total of 15 trucks. Consequently, distinct optimal buffer allocations were obtained for these scenarios, resulting in disparate throughput values. These findings underscore the influence of the number and allocation of trucks on buffer optimisation and subsequent material transfer efficiency in inter-facility operations.

Furthermore, in scenario 4, we observed a near-uniform allocation of buffers across the service stations, except for a single station. This finding underscores the significance of conducting a comprehensive analysis to determine each case's optimal buffer allocation strategy.

Additionally, noteworthy observations can be drawn from scenarios 1 and 2. Despite an increase in the total number of trucks employed (from 9 to 12), the total number of buffers utilised decreased from 22 to 21. This suggests that an increase in truck quantity does not necessarily correlate with a proportional increase in buffer requirements. Similarly, in scenario 4, where the total number of buffers used was 18, an equivalent number of trucks in scenario 5 necessitated 25 buffers to achieve the optimal throughput. This discrepancy highlights the varying impact of truck allocations and compositions on buffer requirements, considering that certain service stations are exclusively utilised by trucks carrying a single product type. In contrast, others are shared among multiple product types.

The decision to set the maximum iterations to 5000 was carefully considered to strike a balance between ensuring convergence towards optimal results and computational efficiency. The figures presented in Appendix A (Figures A1–A4) demonstrate that convergence typically occurs well before reaching the maximum of 5000 iterations, indicating the efficiency of the proposed method in achieving optimal solutions in most experimental examples. However, to account for potential problem size and complexity variations, we adopted a conservative approach by setting a maximum of 5000 iterations. For larger-scale problems, the convergence might occur more slowly than in the experimented examples, making the maximum iteration limit crucial in capturing longer convergence trends. This choice ensures the robustness of our approach, enabling effective handling of a broader range of problem sizes and complexities.

These findings accentuate the importance of considering the interplay between truck allocations, the composition of product types, and their corresponding buffer allocations when optimising material transfer operations. A holistic approach is essential to comprehensively assess and determine each unique scenario's most effective and efficient buffer allocation strategy.

4.1.3. Utilisation Rates Comparison

Efficient utilisation of servers is paramount to optimising system performance and ensuring smooth operations in closed queueing networks. The utilisation rate, representing the ratio of time a server is busy to the total time, is a crucial performance indicator for evaluating server allocation strategies. Table 5 presents the comparison of server utilisation rates obtained through the GEM-based analytical method and simulation model. The results demonstrate the accuracy and reliability of the GEM-based analytical method in estimating server utilisation rates, with average differences between the analytical method and simulation models ranging from 1% to 6% in scenario one and from 1% to 7% in scenario 5, except for nodes 6 and 7, where the utilisation rates are close to 1. However, it should be noted that when utilisation rates approach 1, the GEM-based method may exhibit slightly higher deviations from the simulation results. These findings highlight the effectiveness of the GEM-based method in calculating server utilisation rates in finite closed queueing networks while acknowledging the need for caution when utilisation rates are close to 1.

$$Difference = (Analytical\ meth. - Simulation\ meth.) \tag{11}$$

Table 5. Server utilisation rates in identified scenarios using analytical and simulation method.

Scenario #	**Scenario 1**			**Scenario 5**		
Node #	**GEM Method**	**Simulation Model**	**Diff.**	**GEM Method**	**Simulation Model**	**Diff.**
Node1	63.30%	66%	−2.70%	75.9%	79%	−3.1%
Node2	63.30%	66%	−2.70%	77.8%	78%	−0.2%
Node3	68.90%	66%	2.90%	72.9%	78%	−5.1%
Node4	69.80%	73%	−3.20%	92.4%	99%	−6.6%
Node5	54.10%	55%	−0.90%	54.1%	58%	−3.9%
Node6	85.80%	82%	3.80%	87.5%	98%	−10.5%
Node7	87.80%	84%	3.80%	87.1%	99%	−11.9%
Node8	28.70%	32%	−3.30%	37.5%	44%	−6.5%
Node9	35.80%	30%	5.80%	25.0%	30%	−5.0%
Node10	38.10%	43%	−4.90%	50.0%	51%	−1.0%

Analysing the scenarios provides valuable insights into the system dynamics and potential bottlenecks. Scenario 1 demonstrates a relatively stable system, exhibiting lower utilisation rates than scenario 5. In scenario 5, several nodes approach or reach utilisation rates near 1.0, indicative of bottleneck situations where system performance stagnates. Notably, scenario 5 (18 trucks) exhibits different truck quantities and compositions compared to scenario 1 (9 trucks), which has the fewest number of trucks.

Additionally, consistent observations reveal that the last three nodes consistently exhibit low utilisation rates. This suggests the need for process redesign to balance the workload across both streams rather than exert excessive pressure on certain segments. Potential solutions include:

- Subdividing specific processes into two sub-processes,
- merging idle stations to consolidate workloads, or
- employing additional servers to alleviate the burden on busy servers.

These interventions can contribute to better resource utilisation, reduced congestion, and improved overall system efficiency. Further analysis and targeted optimisation strategies should be considered to address these observed bottlenecks and enhance the performance of the queueing network.

4.1.4. Sensitivity Analysis

Sensitivity analysis is a valuable tool used in decision-making processes to evaluate the influence of various factors on the outcomes of a system or model. Sensitivity analysis provides insights into the robustness and stability of decision-making frameworks by systematically varying these factors and observing the resulting changes in the outputs. In this section, we employ sensitivity analysis techniques to investigate the impact of critical factors on our decision-making model. Through this analysis, we aim to identify the most influential factors, understand their effects on decision outcomes, and enhance the reliability and effectiveness of our decision-making processes.

In this sensitivity analysis section, we examine the variations in utilisation rates, throughput, and buffer allocations resulting from introducing an additional server to node 6 in scenarios 1 and 3. We aim to assess the impact on the system's performance by introducing this additional server. The results of this analysis are presented in Table 6, which provides insights into the changes observed in utilisation rates, throughput, and buffer allocations under these specific conditions. Through this examination, we gain a deeper understanding of how the system responds to introducing an extra server and its implications for overall system performance in scenarios 1 and 3.

Table 6. Impact of adding an extra server in node 6.

Scenario #	Node Utilisation Rate		Throughput		Total # of Buffers	
	1 Server	2 Servers	1 Server	2 Servers	1 Server	2 Servers
Scenario 1	84%	42%	3101	3126	22	10
Scenario 3	99%	67%	3677	4199	24	18

Table 6 unveils intriguing insights regarding the factors influenced by the introduction of an additional server in node 6. In scenario 1, the results demonstrate a slight increase in throughput, reaching 3126 tons, representing a modest improvement of approximately 0.8% compared to the original case. Conversely, scenario 3 exhibits a more substantial boost in throughput, surging to 4199 tons with the inclusion of a dual server in node 6, signifying a significant increase of approximately 14% from the baseline. Furthermore, both scenarios showcase a reduction in the total number of servers utilised and a decrease in the utilisation rate of node 6. These findings emphasise the interconnected nature of design factors within the system, whereby modifications in certain factors influence other performance metrics to varied extents. Consequently, this analysis provides valuable insights for informed decision making when designing system topologies, enabling practitioners to consider the interplay of factors and their implications on system performance.

4.1.5. Comparison of Analytical and Simulation Methodologies

Analytical and simulation methodologies offer unique advantages when comparing methodologies for studying buffer allocations and system performance. Analytical methodologies provide fast and cost-effective analysis, with mathematical equations and formulas enabling quick calculations of system performance measures. They offer closed-form solutions and accessible interpretation, making them suitable for straightforward systems and providing valuable insights. In contrast, simulation methodologies excel at modelling complex and dynamic systems, capturing real-world complexities and uncertainties more accurately. Although more time consuming and resource intensive, simulation models allow for detailed process flows, interactions, and random variations, providing flexibility and realism. The choice between methodologies depends on research goals, system complexity, available resources, and the trade-off between accuracy, time, and cost. Researchers often employ both methods to gain comprehensive insights into buffer allocations and system performance. This integrated approach ensures a balance between efficiency, accuracy, and the ability to capture the intricacies of the studied systems.

4.2. Out-Bound Logistics Process

SM organisation serves two customer types: Standard customers and spot customers. Standard customers, often large manufacturing companies, have long-term contracts, such as contracts of affreightment with SM, ensuring a stable flow of predictable orders. SM can efficiently plan and allocate resources for these known orders, benefiting from established trust and coordination. In contrast, spot customers have short-term or one-time orders, often with short notice and varying requirements. SM must promptly respond to these demands, adapting quickly to their unpredictability. Despite their shorter engagement, spot customers contribute to SM's growth and resource optimisation.

The outbound logistics process for customer trucks carrying steel rebars within the manufacturing company's premises begins with the gate entry process. This involves recording the entry of trucks and initiating necessary security and safety checks. For spot order trucks, an additional process of creditworthiness evaluation is conducted between the gate entry and the order processing process. This evaluation verifies the customer's ability to pay back by assessing their financial stability and credit history. Figure 8 shows the flowchart of the whole outbound logistics process for both standard and spot customers.

After the gate entry and creditworthiness evaluation, the subsequent processes are initiated. Process 2, order processing, involves receiving and verifying the customer's order for steel rebars, confirming pricing and payment details, and generating an order confirmation. The order processing process for standard and spot customers differs in terms of time requirements. The order processing process takes an average amount of time for standard customers whose orders are previously known to the company. Since the company has established relationships with standard customers and has their order details on record, the process can be streamlined and expedited. However, when it comes to spot customers who place orders on short notice or for one-time purchases, the order processing process tends to take longer. The company must allocate additional time for spot customers, as their orders require thorough verification and may require additional documentation and checks. The extended processing time for spot customers is necessary to ensure these special orders' accuracy, compliance, and proper handling.

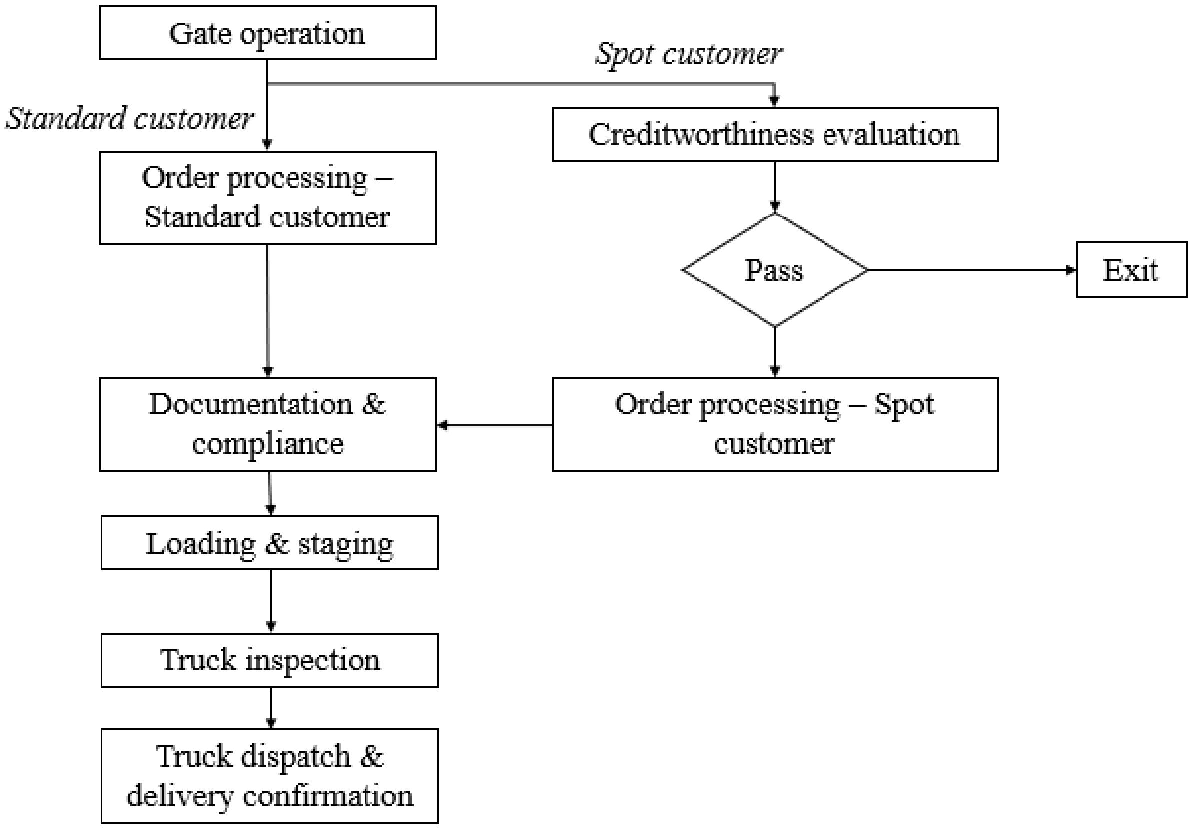


Figure 8. Process flowchart of outbound logistics at SM.

In process 3, steel rebar preparation, the requested steel rebars are retrieved from the inventory or production area within the manufacturing company and checked for compliance with the required specifications. This process includes packaging and labelling and focuses on securely packaging the steel rebars for transport. This may involve bundling the rebars, placing them on pallets, and labelling the packages with relevant information. In process 4, documentation and compliance, the necessary shipping documents, invoices, and paperwork are prepared, ensuring compliance with internal procedures and legal requirements. Process 5, loading and staging, entails loading the packaged steel rebars onto the customer trucks within the manufacturing company's premises. This process includes utilising appropriate equipment, such as forklifts or cranes, to ensure the safe and efficient loading of the rebars onto the trucks. Process 6 involves inspecting the loaded trucks to verify that the steel rebars are adequately secured and that the trucks are in suitable transportation conditions. Any identified concerns or issues are addressed during this truck inspection process.

The truck dispatch process is the final step in outbound logistics, closely coupled with the delivery confirmation. The truck dispatch process comes into play once the trucks carrying steel rebars are loaded and ready to depart from the manufacturing company's premises. Table 7 shows the average service time of sub-processes of outbound logistics operations.

Table 7. Average service time for each outbound logistics processes.

Process	Aver. Service Time (min)	Process	Aver. Service Time (min)
Gate operation	2	Documentation and compliance	2
Creditworthiness evaluation	3	Loading and staging	12
Order processing—standard	7	Truck inspection	2
Order processing—spot	10	Truck dispatch and delivery confirmation	2

4.2.1. DES Model for SM's Outbound Logistics Operation

A DES model, as illustrated in Figure 9, has been constructed to emulate the outbound logistics process of SM. This simulation model incorporates a finite OQN framework within the Anylogic software platform. Subsequently, optimisation configurations are inputted into the model to ascertain the optimal buffer sizes for each service station.

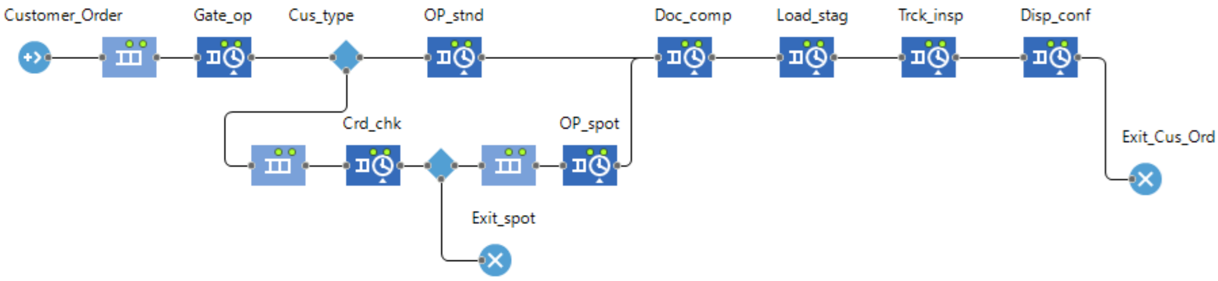


Figure 9. DES model for outbound logistics process.

The simulation-optimisation experiments conducted in this study involve using various customer arrival rates to represent distinct scenarios. The identified customer arrival rates and the ratio between standard and spot customers are presented in Table 8. The objective of the optimisation study is to maximise the throughput, which corresponds to the number of successfully fulfilled customer orders. A constraint is imposed with the budget limit for buffer spaces, with each space costing EUR 100. The total budget allocated for buffer spaces is EUR 8000. The optimisation experiment is executed over 12 h, with a maximum of 1000 iteration runs. These experimental settings align with the methodology employed in the previous section.

Table 8. Identified scenarios for optimisation experiments.

Scenario #	Arrival Rate (per Hour)	Stand.: Spot
Scenario 1	4	80 to 20
Scenario 2	4	90 to 10
Scenario 3	5	90 to 10
Scenario 4	8	90 to 10

4.2.2. Optimal Buffer Allocation for Outbound Logistics Process

Table 9 presents the optimal buffer allocations obtained across the previously identified scenarios. Several noteworthy observations can be drawn from the results presented below. In scenarios 1 and 2, where the arrival rate of trucks remains the same but the customer composition differs, different buffer allocations are observed while maintaining the same throughput. On the other hand, scenario 3 exhibits an optimal uniform buffer allocation. In the final scenario, a higher throughput and lower order processing time are achieved; however, this is accompanied by significantly greater utilisation of buffer spaces. As a result, it is crucial for logistics practitioners in the SM domain to consider various factors such as arrival rate, customer order composition, order processing time, and buffer costs in order to manage their operations effectively.

Table 9. Optimal buffer allocations for identified scenarios with order processing efficiency.

Scenario #	Optimal Buffer Allocation	Throughput (# of Orders Processed)	Aver. Order Processing Time (min)	Total Number of Buffer Spaces Used
Scenario 1	1,1,5,1,7,1,4,3	49	14.69	23
Scenario 2	2,2,2,3,3,1,1,2	49	14.69	16
Scenario 3	1,1,1,1,1,1,1,1	51	14.12	8
Scenario 4	5,12,3,3,20,7,2,16	57	12.63	68

4.2.3. Cycle Time Comparison with Analytical and Simulation Method

Table 10 presents the estimated cycle time (sojourn time) for various scenarios using the analytical method and simulation model. The difference between the results is calculated using Equation (12), ranging between approximately ±4% and ±8%. The table demonstrates that the analytical method, employing the GEM approximation, offers reliable and robust solutions for analysing finite OQNs. The accuracy and effectiveness of the analytical method make it a valuable tool for studying and evaluating system performance in real-world applications.

$$Differnce = \frac{(Analytical\ meth. - Simulation\ meth.)}{Analytical\ meth.} \tag{12}$$

Table 10. Comparison of cycle times using analytical and simulation methods.

Scenario #	Cycle Time (min)		
	GEM Method	**Simulation Model**	**Diff.**
Scenario 1	13.471	14.69	−8.30%
Scenario 2	13.851	14.69	−5.71%
Scenario 3	12.981	14.12	−8.07%
Scenario 4	12.135	12.63	−3.92%

4.3. Managerial Insights

The analysis of buffer allocation in a steel manufacturing company's inter-facility material transfer operation and outbound logistics process reveals several key managerial insights. Firstly, the number and allocation of trucks significantly impact buffer optimisation and material transfer efficiency. By increasing the number of trucks allocated to each material type and the total number of trucks, the overall throughput in total tons of materials transferred can be increased. However, a potential decrease in the average throughput per truck may accompany this increase. Therefore, managers must carefully balance the number of trucks allocated to different material types to achieve optimal performance.

Different optimal buffer allocations were obtained when different truck allocations were implemented for each material type while maintaining a constant total number of trucks, leading to varying throughput values. This finding highlights the importance of considering the quantity and composition of trucks when determining buffer requirements. Furthermore, the analysis reveals that an increase in the total number of trucks does not necessarily result in a proportional increase in buffer requirements. Different scenarios with equivalent numbers of trucks may require significantly different buffer allocations, depending on the truck composition and the specific service stations involved. Thus, managers should consider the specific characteristics of each scenario and carefully analyse the interplay between truck allocations, customer order compositions, and buffer requirements to optimise material transfer operations.

The analysis also emphasises the significance of resource utilisation rates in identifying potential bottlenecks and improving system performance. By evaluating the utilisation rates of servers at various nodes or service stations, managers can pinpoint areas with high levels of occupancy that may hinder system efficiency. In particular, the analysis identifies nodes where utilisation rates approach or reach 1.0, indicating potential bottlenecks. These bottlenecks signal the need for reallocation or additional resources to ensure optimal system performance. Strategies such as process subdivision, workload consolidation, or adding servers can help balance the workload and enhance resource utilisation, reducing congestion and improving overall system efficiency.

Sensitivity analysis further enhances decision making by assessing the impact of critical factors on system outcomes. The analysis demonstrates the influence on throughput, buffer allocations, and utilisation rates by introducing an additional server in specific scenarios. The results indicate that including an extra server can improve throughput and decrease server utilisation, thereby enhancing overall system performance. These findings highlight the system's interconnected nature of design factors and provide valuable insights for informed decision making when designing system topologies.

Overall, the managerial insights derived from the analysis underscore the importance of considering factors such as truck allocations, customer order compositions, buffer requirements, and resource utilisation rates in optimising inter-facility material transfer operations and outbound logistics processes. By carefully evaluating and balancing these factors, managers can effectively manage their operations, improve system performance, and enhance overall efficiency in the steel manufacturing company's supply chain.

5. Conclusions

This paper has presented a framework for optimising buffer allocation in inter-facility material transfer and outbound logistics processes. The study has demonstrated the significance of strategic buffer placement and sizing in improving supply chain efficiency and performance. By utilising finite queueing networks and the generalised expansion method (GEM), the framework allows for the modelling, analysis, and optimisation of buffer allocations in manufacturing systems.

Resource utilisation rates are crucial in identifying bottlenecks and improving system efficiency. Evaluating server utilisation at different nodes allows managers to allocate resources effectively by employing process subdivision or workload consolidation strategies.

This study's significance lies in its pioneering approach to buffer allocation problems, considering both ends of logistics processes. While previous studies focused on single logistics processes, this research integrates inter-facility transfer and outbound logistics, paving the way for a holistic examination of buffer allocation in complex setups. Future research can explore integrated analyses encompassing inbound, intra, and outbound logistics to understand buffer allocation across the entire logistics network comprehensively.

The study has specific limitations that require acknowledgement. Firstly, it concentrates on a single-server environment, overlooking the intricacies of multi-server setups. Secondly, assuming homogeneous jobs or customers may only partially represent real-world situations where variations exist. Thirdly, the study adopts a first-come, first-serve queue discipline, neglecting potential priority-based service considerations. Lastly, the assumption of uniform buffer costs at all stations disregards the possibility of varying costs at different locations. These limitations indicate potential areas for future research to enhance the applicability of buffer allocation models.

In conclusion, this study provides valuable insights into the buffer allocation process for manufacturing companies to optimise inter-facility material transfer and outbound logistics. The proposed framework offers opportunities for improving operational efficiency and overall performance in manufacturing supply chains.

Author Contributions: Conceptualization, M.A. and L.K.; methodology, M.A. and L.K.; software, J.M.S.; validation, L.K., J.M.S. and A.E.; formal analysis, M.A. and L.K.; investigation, L.K. and J.M.S.; resources, L.K. and A.E.; writing—original draft preparation, M.A.; writing—review and editing, L.K. and J.M.S.; visualization, M.A. and A.E.; supervision, L.K. and A.E.; project administration, L.K. and A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

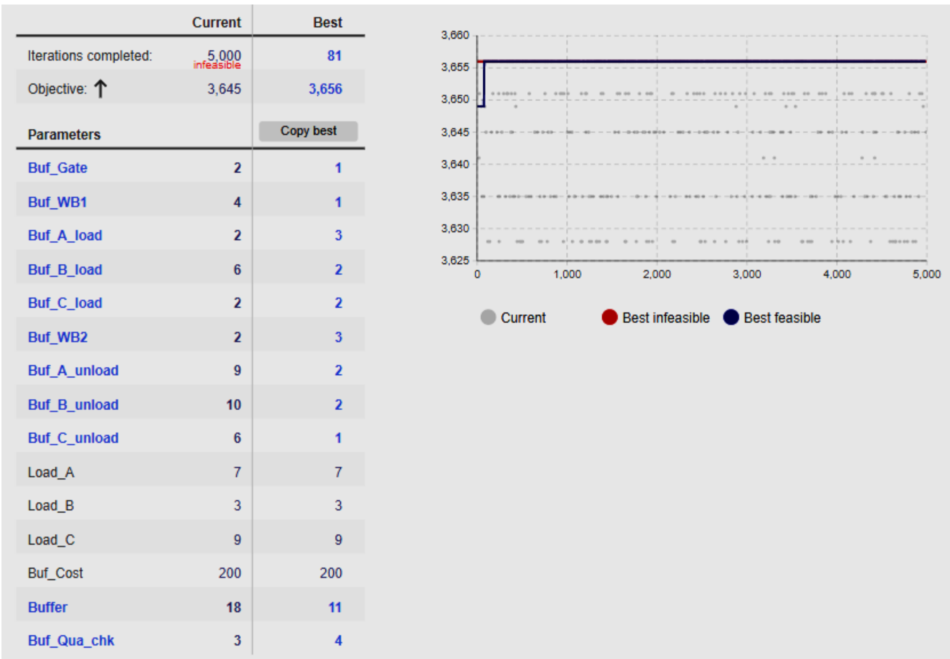
Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Open Access funding provided by the Qatar National Library.

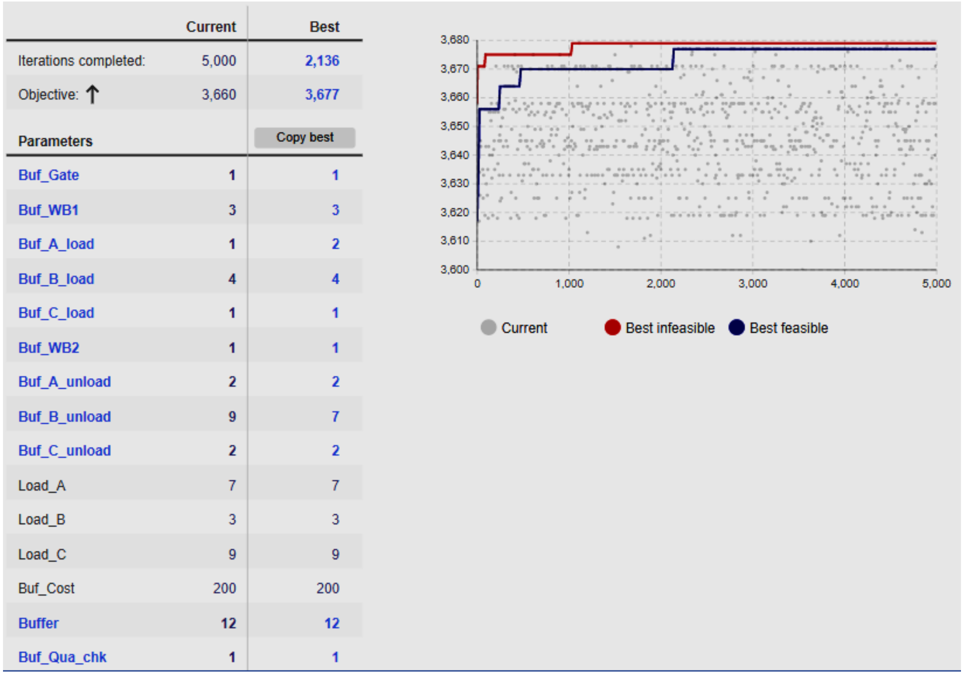
Conflicts of Interest: The authors declare no conflict of interest.

Appendix A



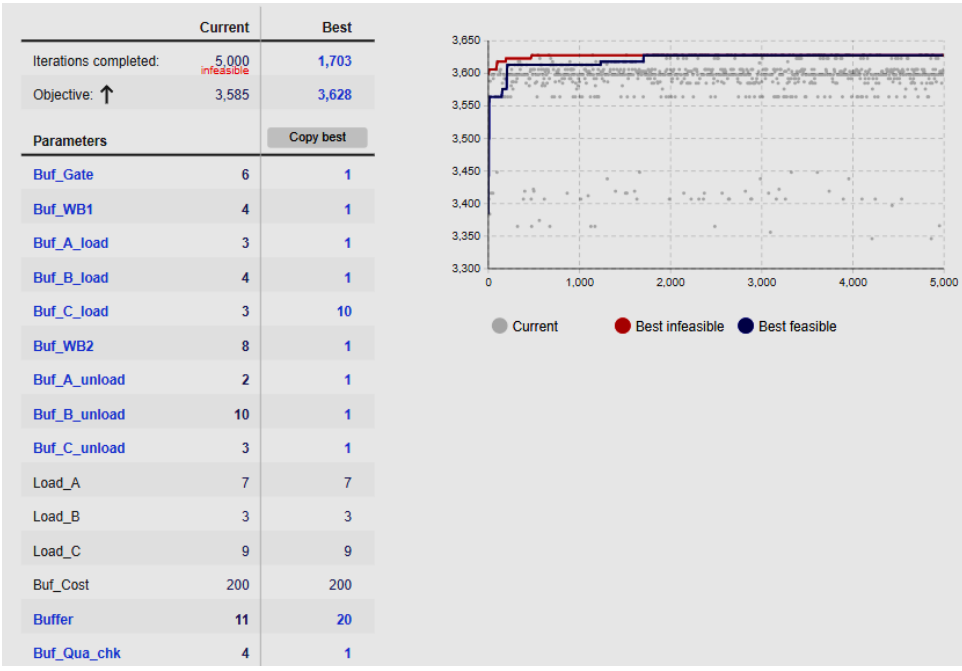
	Current	Best
Iterations completed:	5,000 infeasible	81
Objective: ↑	3,645	3,656
Parameters		Copy best
Buf_Gate	2	1
Buf_WB1	4	1
Buf_A_load	2	3
Buf_B_load	6	2
Buf_C_load	2	2
Buf_WB2	2	3
Buf_A_unload	9	2
Buf_B_unload	10	2
Buf_C_unload	6	1
Load_A	7	7
Load_B	3	3
Load_C	9	9
Buf_Cost	200	200
Buffer	18	11
Buf_Qua_chk	3	4

Figure A1. Optimisation run window—scenario 2.



	Current	Best
Iterations completed:	5,000	2,136
Objective: ↑	3,660	3,677
Parameters		Copy best
Buf_Gate	1	1
Buf_WB1	3	3
Buf_A_load	1	2
Buf_B_load	4	4
Buf_C_load	1	1
Buf_WB2	1	1
Buf_A_unload	2	2
Buf_B_unload	9	7
Buf_C_unload	2	2
Load_A	7	7
Load_B	3	3
Load_C	9	9
Buf_Cost	200	200
Buffer	12	12
Buf_Qua_chk	1	1

Figure A2. Optimisation run window—scenario 3.



	Current	Best
Iterations completed:	5,000 infeasible	1,703
Objective: ↑	3,585	3,628
Parameters		Copy best
Buf_Gate	6	1
Buf_WB1	4	1
Buf_A_load	3	1
Buf_B_load	4	1
Buf_C_load	3	10
Buf_WB2	8	1
Buf_A_unload	2	1
Buf_B_unload	10	1
Buf_C_unload	3	1
Load_A	7	7
Load_B	3	3
Load_C	9	9
Buf_Cost	200	200
Buffer	11	20
Buf_Qua_chk	4	1

Figure A3. Optimisation run window—scenario 4.

Figure A4. Optimisation run window—scenario 5.

References

1. Raja Santhi, A.; Muthuswamy, P. Influence of Blockchain Technology in Manufacturing Supply Chain and Logistics. *Logistics* **2022**, *6*, 15. [CrossRef]
2. Wang, J.; Dou, R.; Muddada, R.R.; Zhang, W. Management of a Holistic Supply Chain Network for Proactive Resilience: Theory and Case Study. *Comput. Ind. Eng.* **2018**, *125*, 668–677. [CrossRef]
3. Chauhan, V.K.; Perera, S.; Brintrup, A. The Relationship between Nested Patterns and the Ripple Effect in Complex Supply Networks. *Int. J. Prod. Res.* **2021**, *59*, 325–341. [CrossRef]
4. Melo, A.C.S.; de Nunes, D.R.L.; Júnior, A.E.B.; de Lima, R.B.; Nagata, V.d.M.N.; Martins, V.W.B. Analysis of Activities That Make up Reverse Logistics Processes: Proposition of a Conceptual Framework. *Braz. J. Oper. Prod. Manag.* **2022**, *19*, e20221376.
5. Handfield, R.B.; Graham, G.; Burns, L. Corona Virus, Tariffs, Trade Wars and Supply Chain Evolutionary Design. *Int. J. Oper. Prod. Manag.* **2020**, *40*, 1649–1660. [CrossRef]
6. MacCarthy, B.L.; Ahmed, W.A.H.; Demirel, G. Mapping the Supply Chain: Why, What and How? *Int. J. Prod. Econ.* **2022**, *250*, 108688. [CrossRef]
7. Beiler, B.C.; Ignácio, P.S.D.A.; Pacagnella Júnior, A.C.; Anholon, R.; Rampasso, I.S. Reverse Logistics System Analysis of a Brazilian Beverage Company: An Exploratory Study. *J. Clean. Prod.* **2020**, *274*, 122624. [CrossRef]
8. Alvarado, C.; Narayanan, L.K.; Shirwaiker, R. Analysis of Tissue Engineered Knee Meniscus Production System Using Discrete Event Simulation. In Proceedings of the 2020 Institute of Industrial and Systems Engineers Annual Conference and Expo, New Orleans, LA, USA, 30 May–2 June 2020; Cromarty, L., Shirwaiker, R., Wang, P., Eds.; Institute of Industrial and Systems Engineers, IISE: Raleigh, NC, USA, 2020; pp. 1527–1532.
9. Gutierrez, L.; Lameijer, B.A.; Anand, G.; Antony, J.; Sunder, M.V. Beyond Efficiency: The Role of Lean Practices and Cultures in Developing Dynamic Capabilities Microfoundations. *Int. J. Oper. Prod. Manag.* **2022**, *42*, 506–536. [CrossRef]
10. Khan, S.A.; Mubarik, M.S.; Kusi-Sarpong, S.; Gupta, H.; Zaman, S.I.; Mubarik, M. Blockchain Technologies as Enablers of Supply Chain Mapping for Sustainable Supply Chains. *Bus. Strateg. Environ.* **2022**, *31*, 3742–3756. [CrossRef]
11. Andry, J.F.; Nurprihatin, F.; Liliana, L. Supply Chain Mapping to Prepare Golden Generation 2045 for Future Technology Infrastructure. In Proceedings of the E3S Web of Conferences, Macau, China, 26–28 April 2022; EDP Sciences: Evry, France, 2022; Volume 359, p. 5004.
12. Queiroz, M.M.; Ivanov, D.; Dolgui, A.; Fosso Wamba, S. Impacts of Epidemic Outbreaks on Supply Chains: Mapping a Research Agenda amid the COVID-19 Pandemic through a Structured Literature Review. *Ann. Oper. Res.* **2022**, *319*, 1159–1196. [CrossRef]
13. Ito, K.; Ikeuchi, K.; Criscuolo, C.; Timmis, J.; Bergeaud, A. Global Value Chains and Domestic Innovation. In *RIETI Discussion Paper 19-E-028, April*; Research Institute of Economy, Trade and Industry: Tokyo, Japan, 2019.
14. Kappel, A.; Schiele, H.; Buchholz, W. Coping with Rising Supply Chain Complexity: Conceptualising a Supply Network Map Structure Model to Address That Challenge. *Int. J. Procure. Manag.* **2020**, *13*, 794–817. [CrossRef]
15. Lafargue, P.; Rogerson, M.; Parry, G.C.; Allainguillaume, J. Broken Chocolate: Biomarkers as a Method for Delivering Cocoa Supply Chain Visibility. *Supply Chain Manag. An Int. J.* **2022**, *27*, 728–741. [CrossRef]

16. Huang, C.Y.; Lee, D.; Chen, S.C.; Tang, W. A Lean Manufacturing Progress Model and Implementation for SMEs in the Metal Products Industry. *Processes* **2022**, *10*, 835. [CrossRef]
17. Sbahieh, S.; Tahir, F.; Al-Ghamdi, S.G. Environmental and Mechanical Performance of Different Fiber Reinforced Polymers in Beams. *Mater. Today Proc.* **2022**, *62*, 3548–3552. [CrossRef]
18. Sbahieh, S.; Mckay, G.; Al-Ghamdi, S.G. A Comparative Life Cycle Assessment of Fiber-Reinforced Polymers as a Sustainable Reinforcement Option in Concrete Beams. *Front. Built Environ.* **2023**, *9*, 1194121. [CrossRef]
19. Kumbhar, M.; Ng, A.H.C.; Bandaru, S. A Digital Twin Based Framework for Detection, Diagnosis, and Improvement of Throughput Bottlenecks. *J. Manuf. Syst.* **2023**, *66*, 92–106. [CrossRef]
20. Cruz, F.R.B.; Smith, J.M.; Queiroz, D.C. Service and Capacity Allocation in M/G/c/c State-Dependent Queueing Networks. *Comput. Oper. Res.* **2005**, *32*, 1545–1563. [CrossRef]
21. Smith, J.M. Robustness of State-Dependent Queues and Material Handling Systems. *Int. J. Prod. Res.* **2010**, *48*, 4631–4663. [CrossRef]
22. Bedell, P.; Smith, J.M. Topological Arrangements of M/G/c/K, M/G/c/c Queues in Transportation and Material Handling Systems. *Comput. Oper. Res.* **2012**, *39*, 2800–2819. [CrossRef]
23. Weiss, S.; Matta, A.; Stolletz, R. Optimization of Buffer Allocations in Flow Lines with Limited Supply. *IISE Trans.* **2018**, *50*, 191–202. [CrossRef]
24. Rodrigues, F.; Agra, A. Berth Allocation and Quay Crane Assignment/Scheduling Problem under Uncertainty: A Survey. *Eur. J. Oper. Res.* **2022**, *303*, 501–524. [CrossRef]
25. Colledani, M.; Tolio, T.; Yemane, A. Production Quality Improvement during Manufacturing Systems Ramp-Up. *CIRP J. Manuf. Sci. Technol.* **2018**, *23*, 197–206. [CrossRef]
26. Kerbache, L.; Smith, J.M.G. Multi-Objective Routing within Large Scale Facilities Using Open Finite Queueing Networks. *Eur. J. Oper. Res.* **2000**, *121*, 105–123. [CrossRef]
27. Romero-Silva, R.; Shaaban, S. Influence of Unbalanced Operation Time Means and Uneven Buffer Allocation on Unreliable Merging Assembly Line Efficiency. *Int. J. Prod. Res.* **2019**, *57*, 1645–1666. [CrossRef]
28. Kłos, S.; Trebuna, P. The Impact of the Availability of Resources, the Allocation of Buffers and Number of Workers on the Effectiveness of an Assembly Manufacturing System. *Manag. Prod. Eng. Rev.* **2017**, *8*, 40–49. [CrossRef]
29. Massim, Y.; Yalaoui, F.; Amodeo, L.; Chatelet, E.; Zeblah, A. Efficient Combined Immune-Decomposition Algorithm for Optimal Buffer Allocation in Production Lines for Throughput and Profit Maximization. *Comput. Oper. Res.* **2010**, *37*, 611–620. [CrossRef]
30. Alden, J.M.; Burns, L.D.; Costy, T.; Hutton, R.D.; Jackson, C.A.; Kim, D.S.; Kohls, K.A.; Owen, J.H.; Turnquist, M.A.; Vander Veen, D.J. General Motors Increases Its Production Throughput. *Interfaces (Providence)* **2006**, *36*, 6–25. [CrossRef]
31. Hemachandra, N.; Eedupuganti, S.K. Performance Analysis and Buffer Allocations in Some Open Assembly Systems. *Comput. Oper. Res.* **2003**, *30*, 695–704. [CrossRef]
32. Eleftheriadis, R.J.; Myklebust, O. A Guideline of Quality Steps towards Zero Defect Manufacturing in Industry. In Proceedings of the IEOM Detroit Conference, IEOM, Detroit, MI, USA, 23–25 September 2016; IEOM Society: Trondheim, Norway, 2016; pp. 332–340.
33. Sharma, A. A Fundamental Study on Qualitatively Viable Sustainable Welding Process Maps. *J. Manuf. Syst.* **2018**, *46*, 221–230. [CrossRef]
34. Araki, M.; Kusakawa, S.; Nakamura, K.; Yonehara, M.; Ikeshoji, T.-T.; Kyogoku, H. Parameter Optimization on the Fabrication of Al-10Si-0.4Mg Alloy Using Selective Laser Melting Process. *Funtai Oyobi Fummatsu Yakin/J. Jpn. Soc. Powder Powder Metall.* **2018**, *65*, 383–388. [CrossRef]
35. Mutua, J.; Nakata, S.; Onda, T.; Chen, Z.-C. Optimization of Selective Laser Melting Parameters and Influence of Post Heat Treatment on Microstructure and Mechanical Properties of Maraging Steel. *Mater. Des.* **2018**, *139*, 486–497. [CrossRef]
36. Bajaj, P.; Wright, J.; Todd, I.; Jägle, E.A. Predictive Process Parameter Selection for Selective Laser Melting Manufacturing: Applications to High Thermal Conductivity Alloys. *Addit. Manuf.* **2019**, *27*, 246–258. [CrossRef]
37. Wang, H.; Zhang, X.; Wang, G.B.; Shen, J.; Zhang, G.Q.; Li, Y.P.; Yan, M. Selective Laser Melting of the Hard-to-Weld IN738LC Superalloy: Efforts to Mitigate Defects and the Resultant Microstructural and Mechanical Properties. *J. Alloys Compd.* **2019**, *807*, 151662. [CrossRef]
38. Ponticelli, G.S.; Guarino, S.; Tagliaferri, V.; Giannini, O. An Optimized Fuzzy-Genetic Algorithm for Metal Foam Manufacturing Process Control. *Int. J. Adv. Manuf. Technol.* **2019**, *101*, 603–614. [CrossRef]
39. He, H.; Cheng, B.; Schmidt, K.; Kruse, C.; Tuffile, C. Numerical Investigation of Extrusion-Based Additive Manufacturing for Process Parameter Planning in a Polymer Dispensing System. In Proceedings of the 30th Annual International Solid Freeform Fabrication Symposium—An Additive Manufacturing Conference, Austin, TX, USA, 12–14 August 2019; The University of Texas at Austin: Chicago, IL, USA, 2019; pp. 1251–1263.
40. Sabzi, H.E.; Maeng, S.; Liang, X.; Simonelli, M.; Aboulkhair, N.T.; Rivera-Díaz-del-Castillo, P.E.J. Controlling Crack Formation and Porosity in Laser Powder Bed Fusion: Alloy Design and Process Optimisation. *Addit. Manuf.* **2020**, *34*, 101360. [CrossRef]
41. Mejri, I.; Ghezal, A.E.; Layeb, S.B. Process Reengineering and Optimization for Innovation Performance: An Application to Detergent Manufacturing. In Proceedings of the 2021 International Conference on Decision Aid Sciences and Application, DASA 2021, Online, 7–8 December 2021; Institute of Electrical and Electronics Engineers Inc.: Tunis, Tunisia, 2021; pp. 220–225.

42. Gui, Y.; Aoyagi, K.; Bian, H.; Chiba, A. Detection, Classification and Prediction of Internal Defects from Surface Morphology Data of Metal Parts Fabricated by Powder Bed Fusion Type Additive Manufacturing Using an Electron Beam. *Addit. Manuf.* **2022**, *54*, 102736. [CrossRef]
43. Akbari, P.; Ogoke, F.; Kao, N.-Y.; Meidani, K.; Yeh, C.-Y.; Lee, W.; Barati Farimani, A. MeltpoolNet: Melt Pool Characteristic Prediction in Metal Additive Manufacturing Using Machine Learning. *Addit. Manuf.* **2022**, *55*, 102817. [CrossRef]
44. Patel, S.; Chen, H.; Vlasea, M.; Zou, Y. The Influence of Beam Focus during Laser Powder Bed Fusion of a High Reflectivity Aluminium Alloy—AlSi10Mg. *Addit. Manuf.* **2022**, *59*, 103112. [CrossRef]
45. Bisogno, S.; Calabrese, A.; Ghiron, N.L.; Pacifici, A. Theory of Constraints Applied to Scheduled and Unscheduled Patient Flows: Does It Improve Process Performance? *Int. J. Serv. Oper. Manag.* **2017**, *26*, 365–385. [CrossRef]
46. Thürer, M.; Stevenson, M. Bottleneck-Oriented Order Release with Shifting Bottlenecks: An Assessment by Simulation. *Int. J. Prod. Econ.* **2018**, *197*, 275–282. [CrossRef]
47. Thürer, M.; Stevenson, M. On the Beat of the Drum: Improving the Flow Shop Performance of the Drum–Buffer–Rope Scheduling Mechanism. *Int. J. Prod. Res.* **2018**, *56*, 3294–3305. [CrossRef]
48. Costa, F.; Kundu, K.; Portioli-Staudacher, A. An Assessment of Order Release Models in Hybrid MTO-MTS Flow Shop with Bottleneck. In Proceedings of the IFIP Advances in Information and Communication Technology, Virtual, 28–29 June 2021; Volume 634, pp. 277–287.
49. Yue, L.; Xu, G.; Mumtaz, J.; Chen, Y.; Zou, T. Order Releasing and Scheduling for a Multi-Item MTO Industry: An Efficient Heuristic Based on Drum Buffer Rope. *Appl. Sci.* **2022**, *12*, 1925. [CrossRef]
50. Saif, U.; Guan, Z.; Wang, C.; He, C.; Yue, L.; Mirza, J. Drum Buffer Rope-Based Heuristic for Multi-Level Rolling Horizon Planning in Mixed Model Production. *Int. J. Prod. Res.* **2019**, *57*, 3864–3891. [CrossRef]
51. Lin, D.; Lee, C.K.M.; Zhou, X. Integration of TOC and ERP in Production Planning. In Proceedings of the 8th International Conference on Logistics, Informatics and Service Sciences, LISS 2018, Toronto, ON, Canada, 3–6 August 2018.
52. Prasetyaningsih, E.; Deferinanda, C.A.; Amaranti, R. Bottleneck Reduction at the Shoes Production Line Using Theory of Constraints Approach. In Proceedings of the ICSECC 2019—International Conference on Sustainable Engineering and Creative Computing, Bandung, Indonesia, 20–22 August 2019; pp. 170–175.
53. Telles, E.S.; Lacerda, D.P.; Morandi, M.I.W.M.; Piran, F.A.S. Drum-Buffer-Rope in an Engineering-to-Order System: An Analysis of an Aerospace Manufacturer Using Data Envelopment Analysis (DEA). *Int. J. Prod. Econ.* **2020**, *222*, 107500. [CrossRef]
54. Liu, M.; Mumtaz, J.; Li, G. Multi-Level Rolling Horizon Planning and Scheduling Integrated with Material Constraints Using DBR Approach: A Heuristic for Smart Manufacturing. In Proceedings of the 2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing, WCMEIM 2022, Ma'anshan, China, 18–20 November 2022; pp. 1192–1197.
55. Cruz, F.R.B.; Van Woensel, T.; Smith, J.M. Buffer and Throughput Trade-Offs in M/G/1/K Queueing Networks: A Bi-Criteria Approach. *Int. J. Prod. Econ.* **2010**, *125*, 224–234. [CrossRef]
56. Smith, J.M.; Cruz, F.R.B.; van Woensel, T. Topological Network Design of General, Finite, Multi-Server Queueing Networks. *Eur. J. Oper. Res.* **2010**, *201*, 427–441. [CrossRef]
57. Wang, G.; Shin, Y.W.; Moon, D.H. Comparison of Three Flow Line Layouts with Unreliable Machines and Profit Maximization. *Flex. Serv. Manuf. J.* **2016**, *28*, 669–693. [CrossRef]
58. Zhang, J.; Qin, W.; Wu, L.H. A Performance Analytical Model of Automated Material Handling System for Semiconductor Wafer Fabrication System. *Int. J. Prod. Res.* **2016**, *54*, 1650–1669. [CrossRef]
59. MacGregor Smith, J. Joint Optimisation of Buffers and Network Population for Closed Finite Queueing Systems. *Int. J. Prod. Res.* **2016**, *54*, 5111–5135. [CrossRef]
60. Li, L.; Qian, Y.; Du, K.; Yang, Y. Analysis of Approximately Balanced Production Lines. *Int. J. Prod. Res.* **2016**, *54*, 647–664. [CrossRef]
61. Cruz, F.R.B.; Duarte, A.R.; Souza, G.L. Multi-Objective Performance Improvements of General Finite Single-Server Queueing Networks. *J. Heuristics* **2018**, *24*, 757–781. [CrossRef]
62. Pedrielli, G.; Matta, A.; Alfieri, A.; Zhang, M. Design and Control of Manufacturing Systems: A Discrete Event Optimisation Methodology. *Int. J. Prod. Res.* **2018**, *56*, 543–564. [CrossRef]
63. Yu, A.-L.; Zhang, H.-Y.; Chen, Q.-X.; Mao, N.; Xi, S.-H. Buffer Allocation in a Flow Shop with Capacitated Batch Transports. *J. Oper. Res. Soc.* **2022**, *73*, 888–904. [CrossRef]
64. Hu, J.; Andradottir, S.; Ayhan, H. Server coordination in queueing systems: When and how? *Probab. Eng. Informational Sci.* **2022**, *36*, 868–922. [CrossRef]
65. Yoon, S.-K.; Jung, K.-S.; Li, X.-S.; Lee, S.-M.; Kim, S.-D. *Hot-Cold Data Filtering and Management for PRAM Based Memory-Storage Unified System*; Institute of Electrical and Electronics Engineers Inc.: Seoul, Republic of Korea, 2017; Volume 2017, pp. 1609–1614.
66. Sousa, T.M.I.; Paz, A.R. How to Evaluate the Quality of Coarse-Resolution DEM-Derived Drainage Networks. *Hydrol. Process.* **2017**, *31*, 3379–3395. [CrossRef]
67. Zhou, X.; Venigalla, M.; Zhu, S. Bounding Box Approach to Network Pruning for Efficient Path Search through Large Networks. *J. Comput. Civ. Eng.* **2017**, *31*, 04017033. [CrossRef]
68. Whitt, W. Towards Better Multi-Class Parametric-Decomposition Approximations for Open Queueing Networks. *Ann. Oper. Res.* **1994**, *48*, 221–248. [CrossRef]

69. Rabta, B. A Review of Decomposition Methods for Open Queueing Networks. *Rapid Model. Increasing Compet. Tools Mindset* **2009**, 25–42. [CrossRef]
70. Hendricks, K.B. *The Steady-State Output Processes of Finite Buffered Production Lines of Identical Machines with Exponential and Non-Exponential Service Times*; Cornell University: Ithaca, NY, USA, 1990; ISBN 9798206261554.
71. Balsamo, S.; de Nitto Personé, V.; Onvural, R. *Analysis of Queueing Networks with Blocking*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; Volume 31, ISBN 0792379969.
72. Kerbache, L.; Smith, J.M. Asymptotic Behavior of the Expansion Method for Open Finite Queueing Networks. *Comput. Oper. Res.* **1988**, *15*, 157–169. [CrossRef]
73. Kerbachea, L.; Smith, J.M. The Generalized Expansion Method for Open Finite Queueing Networks. *Eur. J. Oper. Res.* **1987**, *32*, 448–461. [CrossRef]
74. Kuehn, P. Approximate Analysis of General Queuing Networks by Decomposition. *IEEE Trans. Commun.* **1979**, *27*, 113–126. [CrossRef]
75. Reiser, M.; Kobayashi, H. Accuracy of the Diffusion Approximation for Some Queuing Systems. *IBM J. Res. Dev.* **1974**, *18*, 110–124. [CrossRef]
76. Dorda, M.; Teichmann, D. Modelling of Freight Trains Classification Using Queueing System Subject to Breakdowns. *Math. Probl. Eng.* **2013**, *2013*. [CrossRef]
77. Smith, J.M. *Introduction to Queueing Networks: Theory ∩ Practice*; Springer: Berlin/Heidelberg, Germany, 2018; ISBN 3319788221.
78. Shallal, M.A.; Ali, K.K.; Raslan, K.R.; Rezazadeh, H.; Bekir, A. Exact Solutions of the Conformable Fractional EW and MEW Equations by a New Generalized Expansion Method. *J. Ocean Eng. Sci.* **2020**, *5*, 223–229. [CrossRef]
79. De Souza, G.L.; Duarte, A.R.; Moreira, G.J.P.; Cruz, F.R.B. Post-Processing Improvements in Multi-Objective Optimization of General Single-Server Finite Queueing Networks. *IEEE Lat. Am. Trans.* **2023**, *21*, 381–388. [CrossRef]
80. Cimpoiasu, R. Multiple Explicit Solutions of the 2D Variable Coefficients Chafee-Infante Model via a Generalized Expansion Method. *Mod. Phys. Lett. B* **2021**, *35*, 2150312. [CrossRef]
81. Souza, G.L.; Duarte, A.R.; Moreira, G.J.P.; Cruz, F.R.B. A Novel Formulation for Multi-Objective Optimization of General Finite Single-Server Queueing Networks. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020. [CrossRef]
82. Zhang, H.Y.; Chen, Q.X.; Smith, J.M.G.; Mao, N.; Yu, A.L.; Li, Z.T. Performance Analysis of Open General Queuing Networks with Blocking and Feedback. *Int. J. Prod. Res.* **2017**, *55*, 5760–5781. [CrossRef]
83. Smith, J.M. Simultaneous Buffer and Service Rate Allocation in Open Finite Queueing Networks. *IISE Trans.* **2018**, *50*, 203–216. [CrossRef]
84. Amjath, M.; Kerbache, L.; Macgregor, J.; Elomri, A. Fleet Sizing of Trucks for an Inter-Facility Material Handling System Using Closed Queueing Networks. *Oper. Res. Perspect.* **2022**, *9*, 100245. [CrossRef]
85. The AnyLogic Advanced Modeling with Java | AnyLogic Help. Available online: https://anylogic.help/advanced/index.html (accessed on 20 July 2023).

applied sciences

MDPI

Article

Designing Sustainable Flexible Manufacturing Cells with Multi-Objective Optimization Models †

Emine Bozoklar [1,*] and Ebru Yılmaz [2]

1 Department of Industrial Engineering, Faculty of Engineering and Natural Sciences, İskenderun Technical University, İskenderun, 31200 Hatay, Türkiye
2 Department of Industrial Engineering, Faculty of Engineering, Cukurova University, Sarıçam, 01250 Adana, Türkiye; eyilmaz@cu.edu.tr
* Correspondence: emine.bozoklar@iste.edu.tr
† This study is grounded in, under the supervision of Associate Professor Dr. Ebru YILMAZ, Emine BOZOKLAR's presently ongoing PhD thesis in the Department of Industrial Engineering, Institute of Natural and Applied Sciences, Cukurova University, Sarıçam, 01250 Adana, Türkiye.

Abstract: Having sustainable and flexible features is crucial for manufacturing companies considering the increasing competition in the globalized world. This study considers three aspects of sustainability, namely economic, social, and environmental factors, in the design of flexible manufacturing cells. Three different multi-objective integer mathematical programming models were developed with the objective of minimizing the costs associated with carbon emissions, inter-cellular movements, machine processing, machine replacement, worker training, and additional salary (bonus). Simultaneously, these models aim to minimize the carbon emission amount of the cells within the environmental dimension scope. The developed models are a goal programming model, an epsilon constraint method, and an augmented epsilon constraint (AUGMECON) method. In these models, alternative routes of parts are considered while assigning parts to machines. The results are obtained using the LINGO 20.0 optimization program with a developed illustrative example. The obtained results are tested and compared by performing sensitivity analyses. The sensitivity analyses include examinations of the effects of changes in part demands, machine capacity values, carbon limit value, and the maximum number of workers in cells.

Keywords: sustainable manufacturing; flexible manufacturing cells; multi-objective optimization; goal programming; epsilon constraint method; augmented epsilon constraint method

1. Introduction

Sustainability refers to discussing economic, environmental, and social dimensions simultaneously. While the environmental dimension assesses various factors such as gas emissions, solid/liquid waste management, and energy consumption, the social dimension considers some aspects such as working conditions and work safety. The economic dimension considers providing economic benefits like increasing net present value [1]. Sustainable manufacturing examines products and techniques that have both an economic impact and the ability to minimize the adverse effects of environmental factors, while also protecting energy and natural resources and being reliable for workers [2]. The environmental aspect of sustainability relates to the well-being of people and relies on the responsible use of renewable and non-renewable resources and the Earth's capacity to breathe waste. The environmental aspect of sustainability points out that natural resources are not abundant and are continually consumed. Environmental indicators provide an early warning system to prohibit damage to the natural environment [3]. While manufacturing companies aim to transform natural resources, financial capital, and information into products that fulfill social needs, the human factor plays a crucial role in every aspect of the manufacturing process. Social sustainability indicators are essential in assessing and measuring the social impacts of manufacturing decisions [4]. Galal and Moneim [5] examine social factors within manufacturing systems, addressing elements such as education budget, overtime rate, security, and labor expenditure. In the context of manufacturing system sustainability, Rajak and Vinodh [6] investigate social sustainability indicators, encompassing aspects like job opportunities, health and safety applications, research and development, healthcare and education, and social cohesion. According to Vimal et al. [7], employee education and training emerge as significant strategies for advanced manufacturing systems. Lin et al. [8] categorize social factors in manufacturing systems into diverse criteria, including work accidents, physical workload, working conditions, employee productivity, knowledge and skills, and employee satisfaction. Mengistu and Panizzolo [9] gather a range of criteria, including employment opportunities, employee satisfaction, occupational health and safety, education, and development, and working conditions to define social factors within manufacturing systems. According to Ahmad et al. [3], the economic dimensions of sustainability encompass various indicators that are traditionally associated with financial accounting and intangible assets. The economic dimension of sustainability is indeed interconnected with the environmental and social pillars. Economic pillars are also associated with costs and profits [3].

Sustainable manufacturing systems development efforts contemplate solving problems at all levels (product, process, and system) [10]. The dynamic cellular manufacturing system is one of the manufacturing systems that has a high degree of flexibility and agility to handle product changes [11]. Flexible manufacturing systems are computer-controlled systems that can simultaneously process various parts, including automated material handling equipment and numerically controlled machine tools [12]. The systems divided into subsystems to produce certain parts are called cellular manufacturing systems [13]. The cell formation problem addressing the design of cellular manufacturing systems aims to group parts into part families and related machines into machine cells. This problem aims to ensure grouping efficiency by minimizing inter-cellular and intra-cellular movement costs. The classification of cellular manufacturing systems is based on the geometry of each part and the similarity in the working process. This classification aims to reduce inventory, improve flow time, optimize space utilization, and enhance system efficiency [11].

The organization of this study is as follows: In the second section, a comprehensive review of the related literature is presented, focusing on the environmental, social, and economic dimensions of sustainable factors in cellular manufacturing systems and cell formation. The primary purpose of this study is to design flexible manufacturing cells considering sustainable factors. To achieve this, three different mathematical programming models are presented in the following sections and discussed in Section 3 in detail. The goal programming model, the ε-constraint model, and the augmented ε-constraint (AUGMECON) model were developed. Section 4, Results and Discussion, includes the solution of the sample problem and the sensitivity analyses. In Section 5, the conclusions and future studies are presented.

2. Literature Review

A selection of studies related to the formation of cellular manufacturing systems with some sustainability criteria is given below. Ghodsi et al. [14] consider three aspects of sustainability simultaneously in their cellular manufacturing model. In terms of the economic aspects of sustainability, the lowest cost, the reduction in pollutant emissions as an environmental sustainability criteria, and the reduction in the negative impact on job satisfaction in terms of social factors are taken into consideration. Aljuneidi and Bulgak [15] develop a mixed integer linear programming model that combines reconfigurable cellular manufacturing systems and hybrid manufacturing–remanufacturing systems. They suggest an integrated strategy encompassing aspects of design optimization, analysis, and process planning, aiming to consider several design issues concerning sustainable manufacturing systems. Within the model aiming to minimize the total cost, cost items related to manufacturing and remanufacturing, as well as costs associated with returned products for remanufacturing, were also considered. Mehdizadeh et al. [16] summarize the cell formation problem and production planning and present a multi-objective model. The required time in terms of the time unit for the training of a worker to operate the machines, and some cost terms, such as training, hiring, firing, and salary costs, are considered in the model. Niakan et al. [11] present a two-objective mathematical model for the dynamic cell formation problem considering economic, environmental, and social criteria. In their study, the authors focused on minimizing total cost including cost terms such as inter-cell movement, intra-cell movement, hiring, firing, salary, and training costs, as well as reducing total manufacturing waste, which includes factors such as raw materials, chemicals, energy consumption, and CO_2 emissions. In addition, the maximum daily noise exposure level for worker assignments is added as a constraint to the model as a social criterion. Niakan et al. [17] address minimizing machine-related costs (machine fixed and variable costs, machine procurement and relocation costs, and intra-cell and inter-cell movement costs) and wages, while also considering social issues such as minimizing potential machine hazards and maximizing job opportunities. They state that they tried to establish a balance between economic and social criteria while designing the cells in each period. Imran et al. [18] consider the rated power of machines and the rated power of the material handling devices (AGVs) in the cell formation problem of cellular manufacturing systems. Additionally, the cost per kilowatt hour of electricity is also incorporated into the model. Arghish et al. [19] propose a mathematical model that considers economic and environmental criteria for the type 2 fuzzy cell formation problem. Iqbal and Al-Ghamdi [20] work on saving energy in a machine shop environment by optimizing the assignment of production processes to varied machines and grouping machines in multiple cells to minimize the movement distance. Kumar and Singh [21] propose a bi-objective stochastic mathematical model for sustainable cellular facility layout, along with suggesting an embedded metaheuristic to solve the model. The electricity consumption of AGVs between machines was incorporated into the model. The authors state that the environmental and economic aspects of sustainability in the process of designing a layout is considered in their model. Raoofpanah et al. [22] present a mixed-integer nonlinear programming model that considers environmental issues such as pollution and waste resulting from manufacturing and transportation in the context of cell formation. The cost of the pollution created by the types of vehicles used by the suppliers is considered in the model. Telegraphi and Bulgak [23] present a mixed integer linear programming model for designing optimization of a cellular manufacturing system within the context of a closed-loop supply chain to establish a sustainable manufacturing enterprise. In their study, the minimization of the costs of remanufacturing returned products and related costs such as the disposal, disassembly, and holding of the returned products are also considered. Forghani et al. [24] address an integrated cell formation and group layout model as a mixed-integer program, considering energy consumption, assembly considerations, and process routing. The electric energy consumption generated during processing of parts on machines is incorporated into the model. Jafarzadeh et al. [25] consider the sustainable manufacturing system in the dynamic cellular manufacturing system using fuzzy parameters. They propose a multi-objective sustainable mathematical model that minimizes costs, CO_2 emissions, and product shortages while considering customer satisfaction.

In the literature, various cost items have been considered in relation to cell formation problems. Table 1 provides a chronological presentation of some cost items and various studies that have addressed these costs in the context of cell formation problems.

Table 1. A selection of cost items in cell formation studies.

Authors	Movement Costs		Machine Relocation-Related Cost	Worker Training Cost	Hiring and Firing Cost	Salary Cost	Energy Cost	Remanufacturing Cost	Pollution Cost
	Intra-Cellular	Inter-Cellular							
Aryanezhad et al. [26]		√	√	√	√	√			
Fan and Feng [27]	√	√	√			√			
Bagheri and Bashiri [28]			√	√	√	√			
Aljuneidi and Bulgak [15]			√					√	
Azadeh et al. [29]	√	√	√						
Mehdizadeh and Rahimi [30]				√	√	√			
Mehdizadeh et al. [16]	√	√	√	√	√	√			
Niakan et al. [11]	√	√	√	√	√	√			
Nouri [31]	√	√	√		√	√			
Zohrevand et al. [32]		√	√		√				
Sadeghi et al. [33]	√	√	√	√	√	√			
Zhang and Zhou [34]	√	√				√			
Arghish et al. [19]	√	√					√		
Delgoshaei et al. [35]				√	√	√			
Fahmy [36]	√	√				√			
Raoofpanah et al. [22]	√		√						√

In this study, the design of flexible manufacturing cells is discussed by considering various factors related to economic, social, and environmental dimensions of sustainability. The study aims to design manufacturing cells that quickly adapt to dynamic and competitive market conditions with flexible and sustainable features. A general evaluation of cell formation studies regarding sustainable dimensions is given in Table 2. As seen in Table 2, this study examines the flexible cell formation problem by considering various factors related to sustainability dimensions including economic, environmental, and social dimensions.

Table 2. A selection of cell formation studies according to sustainability dimensions.

Authors	Economical	Environmental	Social
Fan and Feng [27]	√		√
Ghodsi et al. [14]	√	√	√
Aljuneidi and Bulgak [15]	√	√	
Mehdizadeh and Rahimi [30]	√		
Niakan et al. [11]	√	√	√
Niakan et al. [17]	√		√
Nouri [31]	√		
Imran et al. [18]	√		
Sadeghi et al. [33]	√		
Zhang and Zhou [34]	√		√
Arghish et al. [19]	√	√	
Delgoshaei et al. [35]	√		
Iqbal and Al-Ghamdi [20]		√	
Kumar and Singh [21]	√	√	
Raoofpanah et al. [22]	√	√	
Forghani et al. [24]		√	
Jafarzadeh et al. [25]	√	√	√
This article	√	√	√

In Table 3, a selection of various studies in the relevant literature in terms of some factors are listed. This study considers all factors listed in Table 3 and various sustainability factors in the design process of flexible cells.

This study focuses on the design of flexible manufacturing cells considering the economic, environmental, and social dimensions of sustainability. This study aims to ensure optimum results for the following three developed models: the goal programming method, the epsilon constraint (ε-constraint) method, and the AUGMECON method.

Table 3. A selection of cell formation studies in terms of some factors.

Authors	Worker Assignment	Skill	Route Flexibility	Period
Aryanezhad et al. [26]	√	√	√	√
Fan and Feng [27]	√	√	√	√
Bagheri and Bashiri [28]	√			√
Azadeh et al. [29]	√	√		√
Mehdizadeh and Rahimi [30]	√			√
Niakan et al. [11]	√	√		√
Nouri [31]	√		√	√
Sakhaii et al. [37]	√		√	√
Feng et al. [38]	√	√	√	
Raoofpanah et al. [22]				√
Shafiee-Gol et al. [39]			√	√
This article	√	√	√	√

3. Material and Methods

3.1. Problem Formulation

In this article, three different multi-objective mathematical programming models, named the goal programming method, the ε-constraint method, and the AUGMECON method, were developed to address the identified problem. The goal programming model is obtained by minimizing the sum of the deviations from the target values. As mentioned in the study of Felfel et al. [40], in the ε-constraint method, one of the objectives is accepted as the objective function. Thus, while the selected objective function is optimized, other objective functions are considered as constraints and limited by the epsilon value [40]. In the AUGMECON method, unlike the classical ε-constraint method, the slack or surplus variables are included in the model, and the constraints of the objective function are converted into equations [41]. Thus, some constraints that are typical for each multi-objective method were added to the mathematical programming models.

In this study, the developed multi-objective mathematical programming models aim to minimize cost items, including carbon emission, inter-cellular movements, machine processing, machine replacement, worker training, and bonus costs, which are calculated for workers based on their skills. Moreover, this study aims to reduce the amount of carbon emissions by considering the environmental dimension. The optimal route of each part is determined based on alternative routes. In addition, decisions are made to determine the number of machines assigned to cells, added to cells, and removed from cells for each period, the assignment of workers to cells, the number of workers in the cell and system, and the total training time that workers receive. Various assumptions of the developed model are stated below:

- In the system, the routing flexibility for each part is taken into account. Only one alternative route of each part can be chosen.
- The system has machine flexibility. Multiple part types can be processed on different machines.
- The capacities of the cells are limited, and there are upper and lower limits for the number of machines to be taken into account.
- The part demands are fixed and known values. The processing time of a part on its route is known.
- The amount of power that machines consume during production is considered. Indirect and energy-related carbon emissions arising from the operation of the machines are considered. Machine idle times are ignored.
- The movements of parts between cells are considered.
- The carbon emission conversion factor is a constant coefficient.
- The time required for the addition of machines to cells and the removal of machines from the cells is ignored.
- The system has worker flexibility, and different workers may have different skills.
- In addition, worker skills are assumed constant in each period.

3.2. Developed Goal Programming Model

The developed goal programming model consists of several components, including sets, parameters, decision variables, objective function, and constraint equations. These elements are defined as follows:

Indices:	
W	Set of part types ($w \in W$).
R	Set of alternative routes ($r \in R$).
Q	Set of cells ($q \in Q$).
S	Set of machine types ($s \in S$).
I	Set of worker types ($I \in I$).
L	Set of skills ($l \in L$).
E	Set of periods ($e \in E$).
Parameters:	
D_{ew}	The demand of part w in period e.
K_{es}	The time capacity of machine s in period e.
t_{wrqs}	The machining time of part w on machine s in cell q using alternative route r.
pw_{wrqs}	The amount of power consumed while machining part w on machine s in cell q using alternative route r.
HA_{ew}	The cost of moving part w between cells in a period e.
CE_{es}	The machine s carbon emission cost per period e.
O_{es}	The machine s processing/operation cost per period e.
NAC_{es}	The cost of adding machine s to cells per period e.
NRC_{es}	The cost of removing machine s from cells per period e.
EM_{eqi}	The training cost of worker i in cell q in period e.
ES_{il}	The time that worker i spends on an operation of skill l.
HY_{eql}	The limit value of skill l in cell q in period e.
TES_{eql}	The limit time that workers with skill l in cell q in period e can spend.
IK_{eq}	The maximum number of workers of cell q in period e.
HAL_{eq}	The lower bound for the number of machines in cell q in period e.
HUL_{eq}	The upper limit for the number of machines in cell q in period e.
F	The carbon emission conversion factor.
LB_{eq}	The carbon emission limit value of cell q in period e.
B_l	The bonus wage to be received by l skilled worker.
EST_{eil}	The training time received by worker i in the skill l in period e.
z_{ewrq}	$\begin{cases} 1, \text{if part } w \text{ is produced at least one time in cell } q \text{ with alternative route } r \\ \text{in the period } e \\ 0, \text{otherwise} \end{cases}$
IY_{il}	$\begin{cases} 1, \text{if the worker } i \text{ has skill } l \\ 0, \text{otherwise} \end{cases}$
GH_{wrqs}	$\begin{cases} 1, \text{if part } w \text{ has alternative route } r \text{ with machine } s \text{ in cell } q \\ 0, \text{otherwise} \end{cases}$

HD_1, HD_2, HD_3, HD_4, HD_5, HD_6, HD_7, and HD_8, respectively, represent the target values for the objective items.

Decision Variables	
x_{ewr}	$\begin{cases} 1, \text{if alternative route } r \text{ is selected for part } w \text{ in period } e, \\ 0, \text{otherwise} \end{cases}$
V_{eqi}	$\begin{cases} 1, \text{if worker } i \text{ is assigned to cell } q \text{ in period } e, \\ 0, \text{otherwise} \end{cases}$
N_{eqs}	The number of machine s assigned to cell q in period e.
NA_{eqs}	The number of machine s added to cell q in period e.
NR_{eqs}	The number of machine s removed from cell q in period e.
IS_{eq}	The number of workers in cell q in period e.
EIS_e	The total number of workers in the system in period e.
TA_i	The total training time that worker i will receive during all periods according to the worker's abilities.

The parameters K_{es}, t_{wrqs}, ES_{il}, TES_{eql}, and EST_{eil} have the same time unit in the model. Additionally, the parameters HA_{ew}, CE_{es}, O_{es}, NAC_{es}, NRC_{es}, EM_{eqi}, and B_l have the same currency unit.

Objective Function

$$\text{Min} \; = \; d_1^+ + d_2^+ + d_3^+ + d_4^+ + d_5^+ + d_6^+ + d_7^+ + d_8^+ \tag{1}$$

The objective function of the goal programming model is given by Equation (1). Using Equation (1), the minimization of the sum of deviations from the handled targets is ensured. In the objective function (1), $d_1^+, d_2^+, d_3^+, d_4^+, d_5^+, d_6^+, d_7^+, d_8^+$,respectively, are the decision variables that represent positive deviations from the targets. In Equations (2)–(9), $d_1^-, d_2^-, d_3^-, d_4^-, d_5^-, d_6^-, d_7^-, d_8^-$, respectively, are negative deviations from the targets.

Constraints

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs}x_{ewr}GH_{wrqs}D_{ew}pw_{wrqs}F + \left(d_1^- - d_1^+\right) = HD_1 \tag{2}$$

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs}x_{ewr}GH_{wrqs}D_{ew}pw_{wrqs}CE_{es}F + \left(d_2^- - d_2^+\right) = HD_2 \tag{3}$$

The first goal constraint, Equation (2), aims to not exceed the carbon emission target value, as shown by HD_1. Conversion factors can vary according to factors such as fuel types and materials used. In this study, the carbon emissions released from the machines are calculated based on the energy consumption values of the machines. In Equation (3), the second goal constraint of the model, aims to not exceed the target value of the total carbon emission cost. This cost item varies depending on factors such as the power values of the machines, processing times, and part demands.

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs}x_{ewr}N_{eqs}GH_{wrqs}D_{ew}O_{es} + \left(d_3^- - d_3^+\right) = HD_3 \tag{4}$$

Equation (4) aims to not exceed the target value of the total operation cost. The total operation cost is calculated according to factors such as part demands, part processing times, and the number of machines in the cell.

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\left[\left(\sum_{q=1}^{Q} z_{ewrq}\right) - 1\right]D_{ew}HA_{ew}x_{ewr} + \left(d_4^- - d_4^+\right) = HD_4 \tag{5}$$

$$\sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{i=1}^{I} V_{eqi}EM_{eqi} + \left(d_5^- - d_5^+\right) = HD_5 \tag{6}$$

Equation (5), the fourth goal constraint, aims to not pass over the target value of the total cost of movement between cells. The goal constraint indicated by Equation (6) aims to not pass over the target value of the total training cost of workers assigned to cells in each period.

$$\sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{s=1}^{S} NA_{eqs}NAC_{es} + \left(d_6^- - d_6^+\right) = HD_6 \tag{7}$$

$$\sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{s=1}^{S} NR_{eqs}NRC_{es} + \left(d_7^- - d_7^+\right) = HD_7 \tag{8}$$

Equations (7) and (8) are the goal constraints related to the total cost items generated during cell design. Equation (7) aims to not exceed the target value of total cost of the number of machines added to cells. Equation (8) aims to not exceed the target value of the total cost of the item associated with removing machines from cells.

$$\sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{i=1}^{I}\sum_{l=1}^{L} V_{eqi}IY_{il}B_l + \left(d_8^- - d_8^+\right) = HD_8 \tag{9}$$

Equation (9) aims to not exceed the target value of the total bonus wage that workers assigned to cells receive according to their abilities. In this study, the economic dimension of sustainability is also considered when designing the cells.

$$\sum_{r=1}^{R} x_{ewr} = 1 \forall e, w \tag{10}$$

In this study, it is assumed that the model has routing flexibility and each part has alternative routes. Equation (10) shows that the parts can choose only one of their alternative routes in each period.

$$\sum_{w=1}^{W}\sum_{r=1}^{R} z_{ewrq} x_{ewr} \geq 1 \forall e, q \tag{11}$$

$$\sum_{w=1}^{W}\sum_{r=1}^{R} t_{wrqs} x_{ewr} GH_{wrqs} D_{ew} \leq K_{es} N_{eqs} \forall e, q, s \tag{12}$$

Equation (11) shows that at least one part is processed in the selected route in each cell in each period. Equation (12) ensures that the machines cannot exceed their time capacity for each period and each cell.

$$\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{s=1}^{S} t_{wrqs} x_{ewr} GH_{wrqs} D_{ew} pw_{wrqs} F \leq LB_{\mathrm{eq}} \forall e, q \tag{13}$$

Equation (13) shows that the total amount of carbon emissions for each cell in each period cannot exceed a limit value. With this constraint, the environmental dimension of sustainability for the cells is also regarded when designing the cells.

$$N_{e-1,qs} + NA_{eqs} - NR_{eqs} = N_{eqs} \forall e, q, s, e > 1 \tag{14}$$

$$\sum_{s=1}^{S} N_{eqs} \leq HUL_{eq} \forall e, q \tag{15}$$

$$\sum_{s=1}^{S} N_{eqs} \geq HAL_{eq} \forall e, q \tag{16}$$

In Equation (14), regarding cell design, the numbers of machine types in each cell in each period are calculated. The numbers of machine types in each cell are calculated considering that machines may be added to and removed from the cells in each period. Thus, the machine numbers and types change dynamically in each period. Equation (15) shows the upper limit value for the total number of machine types in each cell in each period. In Equation (16), the lower limit value of the total number of machine types for each cell is given.

$$\sum_{i=1}^{I} IY_{il} ES_{il} V_{eqi} \leq TES_{eql} \ \forall e, q, \ l \tag{17}$$

Equation (17) shows that in each period, in each cell and according to each skill, the total time spent by workers cannot exceed a certain limit value. With this constraint, workers are assigned to cells according to their abilities.

$$\sum_{i=1}^{I} V_{eqi} = IS_{eq} \forall e, q \tag{18}$$

$$\sum_{i=1}^{I} V_{eqi} \leq IK_{eq} \forall e, q \tag{19}$$

Equation (18) calculates the total number of workers assigned to each cell in each period. Equation (19) provides that the number of workers assigned to each cell in each period cannot exceed a particular limit value.

$$\sum_{q=1}^{Q} IS_{eq} = EIS_{e} \forall e \tag{20}$$

Equation (20) shows the total number of workers assigned to the cells in each period.

$$\sum_{q=1}^{Q} V_{eqi} \geq 1 \forall e, i \tag{21}$$

$$\sum_{i=1}^{I} IY_{il} V_{eqi} \leq HY_{eql} \forall e, q,\ l \tag{22}$$

$$\sum_{e=1}^{E} \sum_{q=1}^{Q} \sum_{l=1}^{L} V_{eqi} EST_{eil} = TA_i \forall i \tag{23}$$

$$x_{ewr} GH_{wrqs} \leq N_{eqs} \forall e, w, r, q, s \tag{24}$$

Equation (21) provides for the assignment of each worker to at least one cell in each period. Equation (22) shows that the total number of workers for each skill type in each cell for each period cannot exceed the limit value. The total training time received by each worker in all periods is calculated by Equation (23). With Equation (23), the social dimension of sustainability is also regarded when designing the cells. In Equation (24), if the part is produced on its alternative route in a period and the machine type and cell are available with the alternative route of the part, then there is at least one machine assignment to the cell in that period.

$$x_{ewr}, V_{eqi} \in \{0, 1\} \forall e, w, r,\ q, i \tag{25}$$

$$N_{eqs}, NA_{eqs}, NR_{eqs}, EIS_e, IS_{eq} \geq 0 \text{ and integer} \forall e, q, s \tag{26}$$

$$TA_i \geq 0 \forall i \tag{27}$$

$$d_1^+, d_2^+, d_3^+, d_4^+, d_5^+, d_6^+, d_7^+, d_8^+,\ d_1^-, d_2^-, d_3^-, d_4^-, d_5^-, d_6^-, d_7^-, d_8^- \geq 0 \tag{28}$$

0–1 binary decision variables are shown in Equation (25). Decision variables that are positive integers are represented by Equation (26). Equations (27) and (28) indicate that the total training time and the goal deviation values must be positive, respectively.

Linearization of the Model

The goal constraint, indicated by Equation (4), is non-linear due to the multiplication of the two decision variables. For this reason, the constraint in this article is made linear by using the binary-in-integer linearization technique mentioned by Mahdavi et al. [42].

The following new constraint expressions and a decision variable are considered to linearize the model:

$$xn_{ewrqs} = x_{ewr} N_{eqs} \tag{29}$$

$$xn_{ewrqs} \leq N_{eqs} \forall e, w, r, q, s \tag{30}$$

$$xn_{ewrqs} \leq x_{ewr} M\ \forall e, w, r, q, s \tag{31}$$

$$xn_{ewrqs} - N_{eqs} \geq (x_{ewr} - 1) M \forall e, w, r, q, s \tag{32}$$

$$xn_{ewrqs} \geq 0 \text{ and integer} \forall e, w, r, q, s \tag{33}$$

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs} xn_{ewrqs} GH_{wrqs}\ D_{ew} O_{es} + \left(d_3^- - d_3^+\right) = HD_3 \tag{34}$$

M is a big enough number coefficient than the decision variables x_{ewr} and N_{eqs}. The model was made linear with the addition of new Equations (30)–(33), and thus Equation (34) is the edited version of Equation (4). That is, the new Equations (30)–(33) are added to the goal programming model and Equation (34) is added to the model instead of Equation (4).

3.3. Developed ε-Constraint Model

In the ε-constraint method, one of the multi-objective functions is considered as the primary objective function and the other objectives are converted into constraints by applying a limitation with an upper bound. Then, the ε_j level is changed to generate all Pareto solutions [40].

The ε-constraint problem formulation stated in Felfel et al. [40] is taken into account. The below equations are considered for the ε-constraint model in this study. The minimization of total cost items was assigned a relatively higher priority and is considered as a single objective function item in Equation (35).

$$\begin{aligned} MinSNC_2 &= \sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs} x_{ewr} GH_{wrqs} D_{ew} pw_{wrqs} CE_{es} F \\ &+ \sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs} xn_{ewrqs} GH_{wrqs} D_{ew} O_{es} \\ &+ \sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\left[\left(\sum_{q=1}^{Q} z_{ewrq}\right) - 1\right] D_{ew} HA_{ew} x_{ewr} + \sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{i=1}^{I} V_{eqi} EM_{eqi} \\ &+ \sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{s=1}^{S} NA_{eqs} NAC_{es} + \sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{s=1}^{S} NR_{eqs} NRC_{es} \\ &+ \sum_{e=1}^{E}\sum_{q=1}^{Q}\sum_{i=1}^{I}\sum_{l=1}^{L} V_{eqi} IY_{il} B_l \end{aligned} \tag{35}$$

The minimization of the total amount of carbon emissions is represented with SNC_1 and is shown in Equation (36). It was changed to an ε-constraint and thus, Equation (37) is subject to this constraint:

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs} x_{ewr} GH_{wrqs} D_{ew} pw_{wrqs} F = SNC_1 \tag{36}$$

$$\sum_{e=1}^{E}\sum_{w=1}^{W}\sum_{r=1}^{R}\sum_{q=1}^{Q}\sum_{s=1}^{S} t_{wrqs} x_{ewr} GH_{wrqs} D_{ew} pw_{wrqs} F \le \varepsilon_1 \tag{37}$$

Equations (10)–(27), Equations (30)–(33), and Equations (35)–(37) are included in the ε-constraint model in this study.

3.4. Developed Augmented ε-Constraint Model (AUGMECON)

The AUGMECON method is a novel version of the classical ε-constraint method. This method suggests transforming the objective function constraints into equations by including the slack or surplus variables of the classical method [41].

The AUGMECON method improves the classical epsilon constraint method by employing lexicographic optimization to structure the payoff table in alignment with the desired priorities, subsequently optimizing the objective functions in accordance with these priorities. The lexicographic optimization initially focuses on optimizing the first objective function f_1, resulting in the optimal value z_1*. To maintain this optimality for the first objective function, a constraint is introduced, setting f_1 = z_1*, in a model dedicated to optimizing the second objective function. This process is iterated until each individual objective function has been separately optimized. To address another weakness of the epsilon constraint method, the objective function constraints are converted into equalities by introducing slack or surplus variables. The value of δ is a small number [43]. r_i is the range of the i-th objective function, which is determined based on the data in the payoff table [41].

The AUGMECON model formulation stated in Yadollahi et al. [43] was considered. In this study, the below equations are presented. As in the ε-constraint method, minimization of the total cost is considered as relatively important and modeled as a single objective function as shown in Equation (35). The expression SNC_2 represents the total cost and is shown in Equation (35). The minimization of the total amount of carbon emissions is represented with SNC_1. The expression indicated by SLV_1 is a slack or surplus variable in the model.

$$Min(SNC_2 - \delta * (SLV_1/r_1)) \tag{38}$$

$$s.t: \; SNC_1 + SLV_1 = \varepsilon_1 \tag{39}$$

$$SLV_1 \in R^+ \tag{40}$$

In this study, the value of δ is assumed to be 0.0001 as shown in Equation (38). In addition, Equations (10)–(27) and Equations (30)–(33) are included in the AUGMECON model. Additionally, Equations (36) and (38)–(40) are included in the AUGMECON model in this study.

The following Section 4 consists of the results and discussion, where the solution of the sample problem, obtained results, and corresponding sensitivity analyses are presented.

4. Results and Discussion

In this study, a sample problem was created to test the developed mathematical programming model and analyze its sensitivity. The processing time and power consumption of the machines in the cells according to the alternative routes are presented in Table A1 in Appendix A. Table A2 in Appendix A indicates part demands and part movement costs between cells of the sample problem in each planning period.

Figure 1 shows a schematic presentation of the flexible manufacturing cells created using Table A1. In this figure, for example, the flow in the system according to the alternate route 1 of part 1 can be seen.

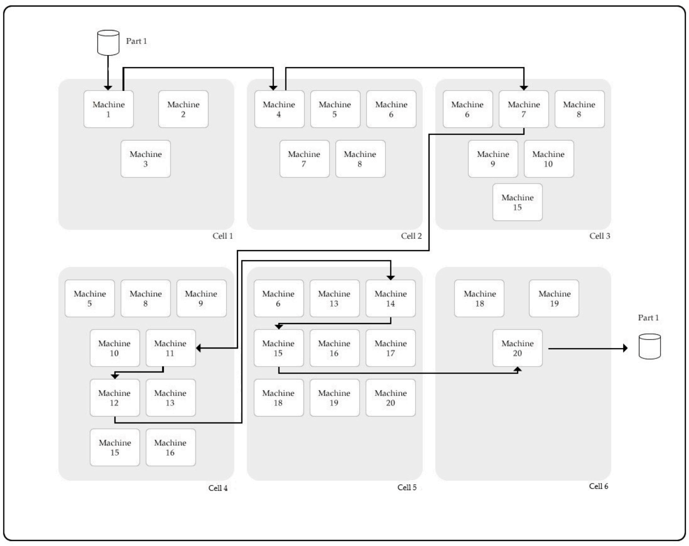


Figure 1. The schematic presentation of the flexible manufacturing cells.

Table 4 shows the time capacities of all machines in each period of the sample problem.

Table 4. Machine time capacities.

Machine	Capacity				
	1. Period	2. Period	3. Period	4. Period	5. Period
1	55,200	51,900	48,800	45,400	35,400
2	48,900	45,400	44,200	43,900	42,900
3	50,100	46,300	44,100	43,300	42,300
4	49,200	48,100	47,100	45,100	44,100
5	48,900	46,800	45,500	44,800	43,800
6	46,200	45,000	44,000	43,500	41,000
7	47,500	45,900	42,000	40,900	38,900
8	44,850	42,750	41,800	40,750	38,750
9	47,200	46,900	45,800	43,900	37,900
10	48,900	44,400	43,700	38,900	35,900
11	51,100	50,300	48,800	45,300	42,900
12	58,200	55,100	53,500	50,400	48,900
13	47,900	45,800	44,800	43,800	41,800
14	48,200	47,000	45,800	45,000	43,900
15	54,500	52,900	50,500	48,900	46,900
16	52,850	50,750	48,200	46,750	41,750
17	48,900	45,800	43,220	42,800	40,800
18	48,200	45,000	43,400	42,000	41,500
19	45,500	43,900	42,000	41,600	40,300
20	47,850	45,750	43,500	42,750	41,750

Table 5 presents the process/operation costs of all machines and carbon emission costs in each period of the sample problem. Table 6 indicates the costs associated with adding machines to the cells and removing machines from the cells in each period.

Table 5. Operation costs of all machines and carbon emission costs.

Machine	Operation Costs					Carbon Emission Costs				
	1. Period	2. Period	3. Period	4. Period	5. Period	1. Period	2. Period	3. Period	4. Period	5. Period
1	9	10	12	13	15	7	8	9	8	12
2	8	10	13	16	17	6	9	10	9	11
3	10	11	14	15	19	7	8	9	9	14
4	11	12	16	18	20	9	7	9	10	13
5	13	13	18	19	23	7	7	7	9	12
6	14	15	16	16	22	6	6	9	8	13
7	15	17	12	13	18	8	7	8	9	12
8	13	15	16	17	19	7	9	9	11	13
9	15	18	19	20	21	7	8	10	12	14
10	9	12	14	18	19	6	9	8	7	13
11	10	13	13	13	19	7	8	6	7	12
12	12	15	19	20	24	9	7	9	8	11
13	12	13	16	17	20	7	7	7	9	12
14	16	18	19	17	18	5	6	10	10	14
15	13	15	16	18	22	8	6	9	10	12
16	12	15	16	17	20	7	9	8	9	10
17	14	18	18	20	23	6	7	9	8	11
18	16	17	21	23	26	6	8	9	8	12
19	13	15	14	17	22	7	6	9	12	13
20	12	16	15	18	20	8	9	9	10	11

Table 6. Machine addition and removal costs.

Machine	Machine Addition Costs					Machine Removal Costs				
	1. Period	2. Period	3. Period	4. Period	5. Period	1. Period	2. Period	3. Period	4. Period	5. Period
1	25	45	30	40	49	55	46	40	45	55
2	55	30	35	38	48	75	55	50	55	58
3	45	35	30	38	51	45	55	50	55	59
4	65	62	45	49	59	65	62	65	75	78
5	48	63	50	59	65	48	75	70	75	79
6	55	55	55	60	63	65	55	55	65	75
7	40	56	50	55	65	60	65	60	65	69
8	50	63	60	60	66	45	50	55	65	70
9	25	45	45	55	58	55	46	40	45	55
10	55	30	35	38	48	75	55	50	55	58
11	45	35	42	43	53	45	55	55	58	65
12	65	62	40	60	65	65	62	60	65	70
13	48	63	45	65	68	48	75	70	73	75
14	55	55	50	50	55	65	55	50	54	60
15	40	56	55	59	63	60	65	60	63	65
16	50	63	60	63	65	45	50	70	74	75
17	48	63	60	65	68	48	75	45	48	55
18	55	55	50	55	65	65	55	55	58	60
19	40	56	55	55	65	60	65	60	65	70
20	50	63	60	65	70	45	50	55	58	70

In Table 7, the minimum and maximum number of machines, the maximum number of workers, and the carbon emission upper limit values of each cell for each period are indicated. In Table 8, the skill types of the workers and the time data pertaining to the amount of time workers spend according to these skill types are shown.

Table 7. Upper and lower limits for cells.

Period	Cell Machine Upper Limit						Cell Machine Lower Limit						Cell Worker Upper Limit						Cell Carbon Emission Upper Limit					
	Cell						Cell						Cell						Cell					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	5	5	4	4	4	5	1	1	1	1	1	1	4	4	4	4	4	4	365,100	376,400	397,500	425,100	376,400	377,500
2	5	5	4	5	5	4	1	1	1	1	1	1	4	4	4	4	4	4	375,000	388,500	387,000	395,000	438,500	377,000
3	5	5	4	5	5	4	1	1	1	1	1	1	4	4	4	4	4	4	385,000	354,000	350,000	405,000	415,000	357,800
4	5	5	4	5	5	4	1	1	1	1	1	1	4	4	4	4	4	4	395,000	364,000	390,000	425,000	435,000	387,800
5	4	4	5	4	5	4	1	1	1	1	1	1	4	4	4	4	4	4	355,000	334,000	370,000	405,000	405,000	357,800

Table 8. Worker–skill matrix and workers' skill durations.

Skill	Worker					
	I1	I2	I3	I4	I5	I6
1	1	0	1	0	0	1
2	1	1	0	0	1	0
3	0	0	1	1	0	0
Skill	**Worker**					
	I1	**I2**	**I3**	**I4**	**I5**	**I6**
1	5	0	2	0	0	3
2	4	8	0	0	7	0
3	0	0	3	9	0	0

The limit times that workers can spend in each cell in each period according to their skill types and training costs are shown in Table 9. Training times received by workers according to their skills are shown in Table 10.

Table 9. The limit values, the limit times of workers with skills, and the training costs.

	The Limit Value of Workers with Skills			**The Limit Time of Workers with Skills**			**Training Cost of Workers**					
	Skill			**Skill**			**Workers**					
1. Period	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	1	2	0	25	20	0	50	65	75	45	35	70
Cell 2	2	2	1	15	30	25	30	80	55	65	35	25
Cell 3	2	0	3	35	0	30	33	65	78	40	25	30
Cell 4	1	2	0	25	20	0	40	75	45	35	35	70
Cell 5	2	2	1	15	30	25	30	80	65	55	35	25
Cell 6	2	0	3	35	0	30	38	65	78	40	25	30
2. Period	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	3	2	0	20	15	0	50	65	75	45	35	70
Cell 2	1	3	0	40	25	0	30	80	55	65	35	25
Cell 3	2	1	2	35	40	25	33	65	78	40	25	30
Cell 4	3	2	0	20	15	0	40	75	45	35	35	70
Cell 5	1	3	0	40	25	0	30	80	65	55	35	25
Cell 6	2	1	2	35	40	25	38	65	78	40	25	30
3. Period	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	3	2	0	20	15	0	50	65	75	45	35	70
Cell 2	1	3	0	40	25	0	30	80	55	65	35	25
Cell 3	2	1	2	35	40	25	33	65	78	40	25	30
Cell 4	3	2	0	20	15	0	40	75	45	35	35	70
Cell 5	1	3	0	40	25	0	30	80	65	55	35	25
Cell 6	2	1	2	55	45	35	38	65	78	40	25	30
4. Period	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	4	3	0	30	35	0	60	75	85	55	45	75
Cell 2	2	3	0	50	35	0	35	70	65	75	55	35
Cell 3	3	2	3	45	40	25	39	65	75	45	35	35
Cell 4	4	2	0	30	35	0	45	78	48	55	45	80
Cell 5	3	3	0	45	35	0	35	85	75	59	45	35
Cell 6	4	2	3	40	50	35	40	75	80	50	30	40
5. Period	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	1	1	0	22	17	0	40	55	70	40	30	65
Cell 2	2	2	1	13	20	20	25	70	50	60	30	20
Cell 3	2	0	3	30	0	25	30	60	70	35	20	25
Cell 4	1	2	0	20	20	0	38	65	40	30	33	68
Cell 5	2	2	1	13	25	27	28	70	60	50	30	23
Cell 6	2	0	2	30	0	25	35	60	75	35	20	28

The carbon emission conversion factor denoted by F is assumed as 0.426 kg/kWh. The bonus wages to be received by the workers according to each skill type are assumed as 900, 700, and 800 currency units, respectively. HD_1, HD_2, HD_3, HD_4, HD_5, HD_6, HD_7, and HD_8 are assumed as 600,000, 610,000, 535,000, 450,000, 8000, 5000, 5000, and 7000, respectively. Additionally, the M value is assumed to be 1000. The GH_{wrqs} parameter is derived based on the t_{wrqs} parameter found in Table A1 in Appendix A. It takes a value of 1 if there is an available machine process time for the alternative route r with machine s in cell q for part w; otherwise, it assumes a value of 0. z_{ewrq} is a parameter that is created according to periods and considers the machine processes in the cells according to the routes of the parts in Table A1 in Appendix A.

In this study, the LINGO 20.0 optimization program was employed to solve the multi-objective integer mathematical programming models, which addresses the design of sustainable and flexible manufacturing cells. The developed goal programming, ε-constraint, and AUGMECON mathematical programming models were solved separately using the LINGO 20.0 optimization program using a MacBook Air (M1, 2020) with 8 GB of RAM. The global optimal results were obtained in 12 min and 38 s, 18 min and 31 s, and 19 min and 51 s for the goal programming method, the ε-constraint method, and the AUGMECON method, respectively. In the results obtained in the goal programming model, the positive and negative deviation values from the related targets are obtained as $d_1^+ = 5{,}330{,}653$, $d_2^+ = 52{,}162{,}290$, $d_3^+ = 14{,}190{,}360$, $d_4^+ = 3{,}317{,}127$, $d_5^+ = 0$, $d_6^+ = 0$, $d_7^+ = 0$, $d_8^+ = 25{,}000$, $d_1^- = 0$, $d_2^- = 0$, $d_3^- = 0$, $d_4^- = 0$, $d_5^- = 6413$, $d_6^- = 4506$, $d_7^- = 4613$, $d_8^- = 0$. Additionally, the objective function value of the goal programming model is obtained as 75,025,430. In Table A3 seen in Appendix B, the results related to optimal routes obtained from these three methods are presented separately. The optimal machine assignments results for the goal programming, ε-constraint, and AUGMECON methods are shown in Tables 11–13, respectively.

Table 10. Training times received by workers according to skills.

	Training Time According to Skill			
	Worker	Skill 1	Skill 2	Skill 3
1. Period	1	15	10	0
	2	0	15	0
	3	17	0	15
	4	0	0	11
	5	0	18	10
	6	18	0	0
	Worker	Skill 1	Skill 2	Skill 3
2. Period	1	10	8	0
	2	0	18	0
	3	17	0	13
	4	0	0	11
	5	0	13	0
	6	17	0	0
	Worker	Skill 1	Skill 2	Skill 3
3.Period	1	10	8	0
	2	0	13	0
	3	15	0	18
	4	0	0	12
	5	0	12	0
	6	23	0	0
	Worker	Skill 1	Skill 2	Skill 3
4.Period	1	15	17	0
	2	0	18	0
	3	22	0	13
	4	0	0	16
	5	0	16	0
	6	20	0	0
	Worker	Skill 1	Skill 2	Skill 3
5.Period	1	11	7	0
	2	0	16	0
	3	15	0	14
	4	0	0	10
	5	0	23	0
	6	15	0	0

Table 11. Optimal machine assignments for the goal programming.

Period	Cell	Optimal Machine Assignments	Optimal Machine Addition	Optimal Machine Removal
1	1	$N_{111}(1)$, $N_{112}(1)$, $N_{113}(1)$		
	2	$N_{124}(1)$, $N_{125}(1)$, $N_{126}(1)$		
	3	$N_{136}(1)$, $N_{137}(1)$, $N_{138}(1)$, $N_{139}(1)$		
	4	$N_{148}(1)$, $N_{1,4,11}(1)$, $N_{1,4,12}(1)$, $N_{1,4,15}(1)$		
	5	$N_{1,5,14}(1)$, $N_{1,5,15}(1)$, $N_{1,5,16}(1)$, $N_{1,5,17}(1)$		
	6	$N_{1,6,18}(1)$, $N_{1,6,19}(1)$, $N_{1,6,20}(1)$		
2	1	$N_{211}(1)$, $N_{212}(1)$, $N_{213}(1)$		
	2	$N_{224}(1)$, $N_{225}(1)$, $N_{226}(1)$		
	3	$N_{236}(1)$, $N_{237}(1)$, $N_{238}(1)$, $N_{239}(1)$		
	4	$N_{2,4,10}(1)$, $N_{2,4,11}(1)$, $N_{2,4,12}(1)$, $N_{2,4,13}(1)$, $N_{2,4,15}(1)$	$NA_{2,4,10}(1)$ $NA_{2,4,13}(1)$	$NR_{248}(1)$
	5	$N_{2,5,14}(1)$, $N_{2,5,15}(1)$, $N_{2,5,16}(1)$, $N_{2,5,17}(1)$		
	6	$N_{2,6,18}(1)$, $N_{2,6,19}(1)$, $N_{2,6,20}(1)$		
3	1	$N_{311}(1)$, $N_{312}(1)$, $N_{313}(1)$		
	2	$N_{324}(1)$, $N_{325}(1)$, $N_{326}(1)$		
	3	$N_{336}(1)$, $N_{337}(1)$, $N_{338}(1)$, $N_{339}(1)$		
	4	$N_{3,4,8}(1)$, $N_{3,4,10}(1)$, $N_{3,4,11}(1)$, $N_{3,4,12}(1)$, $N_{3,4,15}(1)$	$NA_{348}(1)$	$NR_{3,4,13}(1)$
	5	$N_{3,5,14}(1)$, $N_{3,5,15}(1)$, $N_{3,5,16}(1)$, $N_{3,5,17}(1)$		
	6	$N_{3,6,18}(1)$, $N_{3,6,19}(1)$, $N_{3,6,20}(1)$	$NA_{3,6,13}(1)$	
4	1	$N_{411}(1)$, $N_{412}(1)$, $N_{413}(1)$		
	2	$N_{424}(1)$, $N_{425}(1)$, $N_{426}(1)$, $N_{427}(1)$	$NA_{427}(1)$	
	3	$N_{436}(1)$, $N_{437}(1)$, $N_{438}(1)$, $N_{439}(1)$		
	4	$N_{4,4,10}(1)$, $N_{4,4,11}(1)$, $N_{4,4,12}(1)$, $N_{4,4,13}(1)$, $N_{4,4,15}(1)$	$NA_{4,4,13}(1)$	$NR_{448}(1)$
	5	$N_{4,5,14}(1)$, $N_{4,5,15}(1)$, $N_{4,5,16}(1)$, $N_{4,5,17}(1)$, $N_{4,5,18}(1)$	$NA_{4,5,18}(1)$	
	6	$N_{4,6,18}(1)$, $N_{4,6,19}(1)$, $N_{4,6,20}(1)$		
5	1	$N_{511}(1)$, $N_{512}(1)$, $N_{513}(1)$, $N_{5,1,14}(1)$	$NA_{5,1,14}(1)$	
	2	$N_{524}(1)$, $N_{525}(1)$, $N_{526}(1)$		$NR_{527}(1)$
	3	$N_{536}(1)$, $N_{537}(1)$, $N_{538}(1)$, $N_{539}(1)$		
	4	$N_{548}(1)$, $N_{5,4,11}(1)$, $N_{5,4,12}(1)$, $N_{5,4,15}(1)$	$NA_{548}(1)$	$NR_{5,4,10}(1)$ $NR_{5,4,13}(1)$
	5	$N_{5,5,14}(1)$, $N_{5,5,15}(1)$, $N_{5,5,16}(1)$, $N_{5,5,17}(1)$, $N_{5,5,18}(1)$		
	6	$N_{5,6,13}(1)$, $N_{5,6,18}(1)$, $N_{5,6,19}(1)$, $N_{5,6,20}(1)$		

Table 12. Optimal machine assignments for the ε-constraint method.

Period	Cell	Optimal Machine Assignments	Optimal Machine Addition	Optimal Machine Removal
1	1	$N_{111}(1)$, $N_{112}(1)$, $N_{113}(1)$		
	2	$N_{124}(1)$, $N_{125}(1)$, $N_{126}(1)$, $N_{127}(1)$		
	3	$N_{136}(1)$, $N_{137}(1)$, $N_{138}(1)$, $N_{139}(1)$		
	4	$N_{1,4,11}(1)$, $N_{1,4,12}(1)$, $N_{1,4,13}(1)$, $N_{1,4,15}(1)$		
	5	$N_{1,5,14}(1)$, $N_{1,5,15}(1)$, $N_{1,5,16}(1)$, $N_{1,5,17}(1)$		
	6	$N_{1,6,18}(1)$, $N_{1,6,19}(1)$, $N_{1,6,20}(1)$		
2	1	$N_{211}(1)$, $N_{212}(1)$, $N_{213}(1)$		
	2	$N_{224}(1)$, $N_{225}(1)$, $N_{226}(1)$, $N_{227}(1)$		
	3	$N_{236}(1)$, $N_{237}(1)$, $N_{238}(1)$, $N_{239}(1)$		
	4	$N_{2,4,10}(1)$, $N_{2,4,11}(1)$, $N_{2,4,12}(1)$, $N_{2,4,13}(1)$, $N_{2,4,15}(1)$	$NA_{2,4,10}(1)$	
	5	$N_{2,5,14}(1)$, $N_{2,5,15}(1)$, $N_{2,5,16}(1)$, $N_{2,5,17}(1)$		
	6	$N_{2,6,18}(1)$, $N_{2,6,19}(1)$, $N_{2,6,20}(1)$		
3	1	$N_{311}(1)$, $N_{312}(1)$, $N_{313}(1)$		
	2	$N_{324}(1)$, $N_{325}(1)$, $N_{326}(1)$, $N_{327}(1)$		
	3	$N_{336}(1)$, $N_{337}(1)$, $N_{338}(1)$, $N_{339}(1)$		
	4	$N_{3,4,8}(1)$, $N_{3,4,10}(1)$, $N_{3,4,11}(1)$, $N_{3,4,12}(1)$, $N_{3,4,15}(1)$	$NA_{3,4,8}(1)$	$NR_{3,4,13}(1)$
	5	$N_{3,5,14}(1)$, $N_{3,5,15}(1)$, $N_{3,5,16}(1)$, $N_{3,5,17}(1)$		
	6	$N_{3,6,18}(1)$, $N_{3,6,19}(1)$, $N_{3,6,20}(1)$		
4	1	$N_{411}(1)$, $N_{412}(1)$, $N_{413}(1)$		
	2	$N_{424}(1)$, $N_{425}(1)$, $N_{426}(1)$, $N_{427}(1)$		
	3	$N_{436}(1)$, $N_{437}(1)$, $N_{438}(1)$, $N_{439}(1)$		
	4	$N_{4,4,10}(1)$, $N_{4,4,11}(1)$, $N_{4,4,12}(1)$, $N_{4,4,13}(1)$, $N_{4,4,15}(1)$	$NA_{4,4,13}(1)$	$NR_{4,4,8}(1)$
	5	$N_{4,5,14}(1)$, $N_{4,5,15}(1)$, $N_{4,5,16}(1)$, $N_{4,5,17}(1)$		
	6	$N_{4,6,18}(1)$, $N_{4,6,19}(1)$, $N_{4,6,20}(1)$		
5	1	$N_{511}(1)$, $N_{512}(1)$, $N_{513}(1)$		
	2	$N_{524}(1)$, $N_{525}(1)$, $N_{526}(1)$, $N_{527}(1)$		
	3	$N_{536}(1)$, $N_{537}(1)$, $N_{538}(1)$, $N_{539}(1)$		
	4	$N_{548}(1)$, $N_{5,4,11}(1)$, $N_{5,4,12}(1)$, $N_{5,4,15}(1)$	$NA_{548}(1)$	$NR_{5,4,10}(1)$ $NR_{5,4,13}(1)$
	5	$N_{5,5,14}(1)$, $N_{5,5,15}(1)$, $N_{5,5,16}(1)$, $N_{5,5,17}(1)$		
	6	$N_{5,6,18}(1)$, $N_{5,6,19}(1)$, $N_{5,6,20}(1)$		

Table 13. Optimal machine assignments for the AUGMECON method.

Period	Cell	Optimal Machine Assignments	Optimal Machine Addition	Optimal Machine Removal
1	1	$N_{111}(1)$, $N_{112}(1)$, $N_{113}(1)$		
	2	$N_{124}(1)$, $N_{125}(1)$, $N_{126}(1)$		
	3	$N_{136}(1)$, $N_{137}(1)$, $N_{138}(1)$, $N_{139}(1)$		
	4	$N_{1,4,11}(1)$, $N_{1,4,12}(1)$, $N_{1,4,13}(1)$, $N_{1,4,15}(1)$		
	5	$N_{1,5,14}(1)$, $N_{1,5,15}(1)$, $N_{1,5,16}(1)$, $N_{1,5,17}(1)$		
	6	$N_{1,6,18}(1)$, $N_{1,6,19}(1)$, $N_{1,6,20}(1)$		
2	1	$N_{211}(1)$, $N_{212}(1)$, $N_{213}(1)$		
	2	$N_{224}(1)$, $N_{225}(1)$, $N_{226}(1)$		
	3	$N_{236}(1)$, $N_{237}(1)$, $N_{238}(1)$, $N_{239}(1)$		
	4	$N_{2,4,10}(1)$, $N_{2,4,11}(1)$, $N_{2,4,12}(1)$, $N_{2,4,13}(1)$, $N_{2,4,15}(1)$	$NA_{2,4,10}(1)$	
	5	$N_{2,5,14}(1)$, $N_{2,5,15}(1)$, $N_{2,5,16}(1)$, $N_{2,5,17}(1)$		
	6	$N_{2,6,18}(1)$, $N_{2,6,19}(1)$, $N_{2,6,20}(1)$		
3	1	$N_{311}(1)$, $N_{312}(1)$, $N_{313}(1)$		
	2	$N_{324}(1)$, $N_{325}(1)$, $N_{326}(1)$		
	3	$N_{336}(1)$, $N_{337}(1)$, $N_{338}(1)$, $N_{339}(1)$		
	4	$N_{3,4,8}(1)$, $N_{3,4,10}(1)$, $N_{3,4,11}(1)$, $N_{3,4,12}(1)$, $N_{3,4,15}(1)$	$NA_{3,4,8}(1)$	$NR_{3,4,13}(1)$
	5	$N_{3,5,14}(1)$, $N_{3,5,15}(1)$, $N_{3,5,16}(1)$, $N_{3,5,17}(1)$		
	6	$N_{3,6,18}(1)$, $N_{3,6,19}(1)$, $N_{3,6,20}(1)$		
4	1	$N_{411}(1)$, $N_{412}(1)$, $N_{413}(1)$		
	2	$N_{424}(1)$, $N_{425}(1)$, $N_{426}(1)$		
	3	$N_{436}(1)$, $N_{437}(1)$, $N_{438}(1)$, $N_{439}(1)$		
	4	$N_{4,4,8}(1)$, $N_{4,4,10}(1)$, $N_{4,4,11}(1)$, $N_{4,4,12}(1)$, $N_{4,4,15}(1)$		
	5	$N_{4,5,14}(1)$, $N_{4,5,15}(1)$, $N_{4,5,16}(1)$, $N_{4,5,17}(1)$		
	6	$N_{4,6,18}(1)$, $N_{4,6,19}(1)$, $N_{4,6,20}(1)$		
5	1	$N_{511}(1)$, $N_{512}(1)$, $N_{513}(1)$		
	2	$N_{524}(1)$, $N_{525}(1)$, $N_{526}(1)$		
	3	$N_{536}(1)$, $N_{537}(1)$, $N_{538}(1)$, $N_{539}(1)$		
	4	$N_{5,4,8}(1)$, $N_{5,4,11}(1)$, $N_{5,4,12}(1)$, $N_{5,4,15}(1)$		$NR_{5,4,10}(1)$
	5	$N_{5,5,14}(1)$, $N_{5,5,15}(1)$, $N_{5,5,16}(1)$, $N_{5,5,17}(1)$		
	6	$N_{5,6,18}(1)$, $N_{5,6,19}(1)$, $N_{5,6,20}(1)$		

The optimal worker assignments for multi-objective approaches are shown in Table 14. In Table 15, the number of workers in cell q in the period e is shown for each multi-objective approach. Moreover, for each multi-objective approach it was determined that $EIS_1 = 6$, $EIS_2 = 6$, $EIS_3 = 6$, $EIS_4 = 6$, $EIS_5 = 6$, $TA_1 = 111$, $TA_2 = 80$, $TA_3 = 159$, $TA_4 = 60$, $TA_5 = 82$, and $TA_6 = 93$.

Table 14. Optimal worker assignments to cells for each period for the multi-objective approaches.

Period	Optimal Worker Assignment for Goal Programming	Optimal Worker Assignment for ε-Constraint	Optimal Worker Assignment for AUGMECON
1	V_{111}, V_{112}, V_{146}, V_{155}, V_{163}, V_{164}	V_{112}, V_{121}, V_{123},V_{155}, V_{156}, V_{164}	V_{112}, V_{121}, V_{123},V_{155}, V_{156}, V_{164}
2	V_{241}, V_{233},V_{234}, V_{246}, V_{252}, V_{255}	V_{226}, V_{232}, V_{233}, V_{234}, V_{251}, V_{265}	V_{226}, V_{232}, V_{233}, V_{234}, V_{251}, V_{265}
3	V_{333}, V_{334}, V_{341}, V_{346}, V_{352}, V_{355}	V_{326}, V_{332}, V_{333}, V_{334}, V_{351}, V_{365}	V_{326}, V_{332}, V_{333}, V_{334}, V_{351}, V_{365}
4	V_{421}, V_{422}, V_{425}, V_{426}, V_{433}, V_{434}	V_{421}, V_{426}, V_{432}, V_{433}, V_{434}, V_{465}	V_{421}, V_{426}, V_{432}, V_{433}, V_{434}, V_{465}
5	V_{512}, V_{521}, V_{525}, V_{563}, V_{564}, V_{566}	V_{512}, V_{521}, V_{523}, V_{525}, V_{556}, V_{564}	V_{512}, V_{521}, V_{523}, V_{525}, V_{556}, V_{564}

Table 15. Optimal number of workers in cells for each period for the multi-objective approaches.

Period	Optimal Total Worker for Goal Programming	Optimal Total Worker for ε-Constraint	Optimal Total Worker for AUGMECON
1	$IS_{11}(2)$, $IS_{12}(0)$, $IS_{13}(0)$, $IS_{14}(1)$, $IS_{15}(1)$, $IS_{16}(2)$	$IS_{11}(1)$, $IS_{12}(2)$, $IS_{13}(0)$, $IS_{14}(0)$, $IS_{15}(2)$, $IS_{16}(1)$	$IS_{11}(1)$, $IS_{12}(2)$, $IS_{13}(0)$, $IS_{14}(0)$, $IS_{15}(2)$, $IS_{16}(1)$
2	$IS_{21}(0)$, $IS_{22}(0)$, $IS_{23}(2)$, $IS_{24}(2)$, $IS_{25}(2)$, $IS_{26}(0)$	$IS_{21}(0)$, $IS_{22}(1)$, $IS_{23}(3)$, $IS_{24}(0)$, $IS_{25}(1)$, $IS_{26}(1)$	$IS_{21}(0)$, $IS_{22}(1)$, $IS_{23}(3)$, $IS_{24}(0)$, $IS_{25}(1)$, $IS_{26}(1)$
3	$IS_{31}(0)$, $IS_{32}(0)$, $IS_{33}(2)$, $IS_{34}(2)$, $IS_{35}(2)$, $IS_{36}(0)$	$IS_{31}(0)$, $IS_{32}(1)$, $IS_{33}(3)$, $IS_{34}(0)$, $IS_{35}(1)$, $IS_{36}(1)$	$IS_{31}(0)$, $IS_{32}(1)$, $IS_{33}(3)$, $IS_{34}(0)$, $IS_{35}(1)$, $IS_{36}(1)$
4	$IS_{41}(0)$, $IS_{42}(4)$, $IS_{43}(2)$, $IS_{44}(0)$, $IS_{45}(0)$ $IS_{46}(0)$	$IS_{41}(0)$, $IS_{42}(2)$, $IS_{43}(3)$, $IS_{44}(0)$, $IS_{45}(0)$, $IS_{46}(1)$	$IS_{41}(0)$, $IS_{42}(2)$, $IS_{43}(3)$, $IS_{44}(0)$, $IS_{45}(0)$, $IS_{46}(1)$
5	$IS_{51}(1)$, $IS_{52}(2)$, $IS_{53}(0)$, $IS_{54}(0)$, $IS_{55}(0)$ $IS_{56}(3)$	$IS_{51}(1)$, $IS_{52}(3)$, $IS_{53}(0)$, $IS_{54}(0)$, $IS_{55}(1)$, $IS_{56}(1)$	$IS_{51}(1)$, $IS_{52}(3)$, $IS_{53}(0)$, $IS_{54}(0)$, $IS_{55}(1)$, $IS_{56}(1)$

When calculating ε values, the minimum objective function values were obtained for each objective function; hence, the calculated pay-off table for the ε-constraint method is shown in Table 16. The epsilon values for the ε-constraint method are taken between these ε_1 = 5,958,372, . . ., 5,910,828 ranges. The calculated lexicographic optimization pay-off table for the AUGMECON method is shown in Table 17. In the AUGMECON model, the value of r_1 is 45,157. The epsilon values for the AUGMECON method are taken between these ε_1= 5,966,694, . . ., 5,929,064 ranges.

Table 16. Pay-off table for the ε-constraint method.

	SNC_1	SNC_2
Min SNC_1	**5,902,903**	72,492,480
Min SNC_2	5,974,220	**71,788,790**

Table 17. Pay-off table with the lexicographic optimization for the AUGMECON method.

	SNC_1	SNC_2
Min SNC_1	**5,929,063**	**71,788,790**
Min SNC_2	5,974,220	**71,788,790**

The Pareto optimal front graph obtained by using epsilon values in the ε-constraint method is presented in Figure 2.

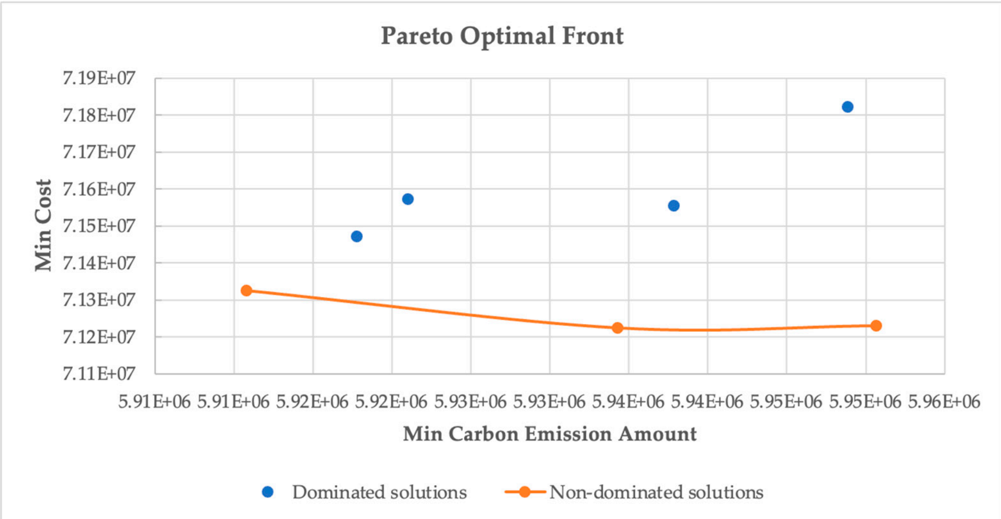


Figure 2. Pareto front for the ε-constraint method.

The following section presents the obtained results from the analyses.

Sensitivity Analyses

Sensitivity analyses were conducted to evaluate the impact of some parameters on the objective function value in the sample problem. The analyses were performed for the three developed multi-objective models: the goal programming, the ε-constraint, and the AUGMECON methods. Firstly, a sensitivity analysis is conducted using the goal programming method to examine the impact of changes in part demands. The results of the analysis are depicted in Figure 3, which illustrates the effect of a 10% decrease in demand for each period individually. For instance, in the first period, the demand value for part 1 is 150, and thus, for the purpose of this analysis, the demand for part 1 is considered as 135. Similarly, the analysis considers a 10% decrease in demand for each part in every period. The impact of percentage changes in part demand values on the cost items of the objective function was evaluated using the goal programming method, and the results are presented in Figure 4. When there is an increase in demand for parts, the costs related to carbon emission, operation, inter-cellular movement, and adding and removing machines are higher than in their current situations. It can be observed in Figure 4 that an increase in part demand does not cause any change in worker training cost and bonus wage items. Raoofpanah et al. [22] examine the effect of changes in demand on costs related to cell formation, inventory, and environmental issues. They state that cell formation costs are more sensitive to changes in demand compared with the other two costs.

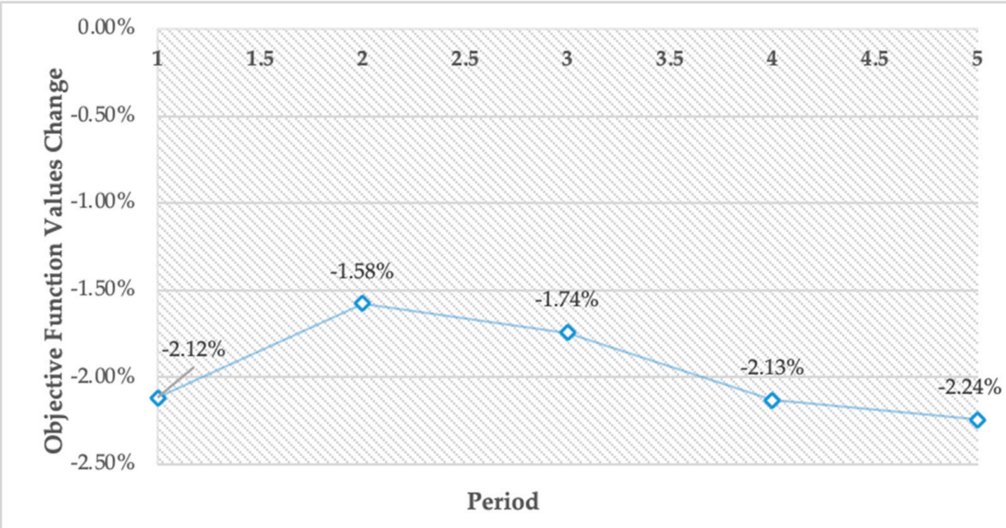


Figure 3. The impact of part demand changes using the goal programming method.

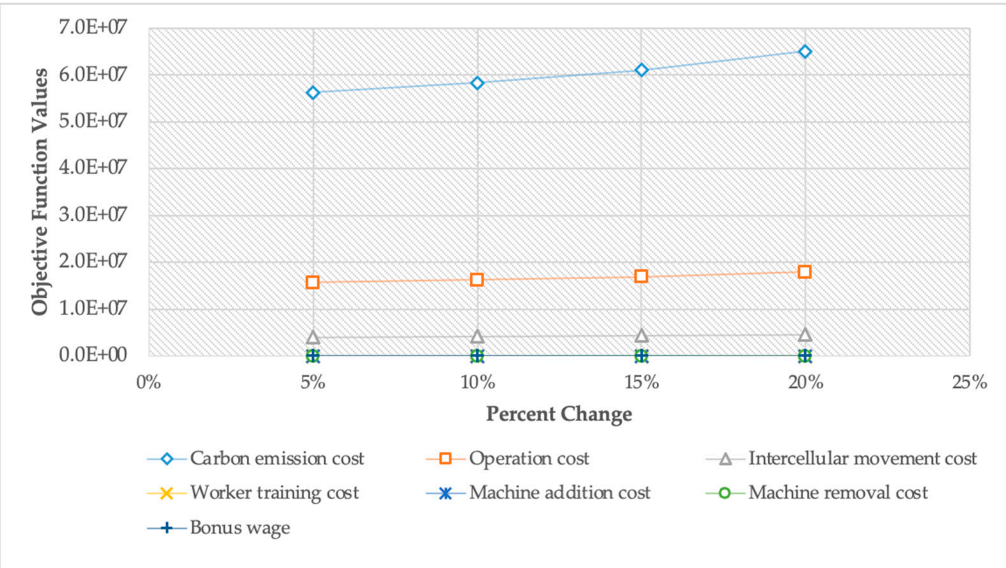


Figure 4. The effect of the increase in demand for parts on the values of cost items using the goal programming method.

Changes in the capacity values of the machines may affect the objective function values of the model. For example, Figure 5 illustrates the analysis of the impact of a 10% increase in capacity value of machine 1 in period 5 on both the carbon emission amount and total cost, which are objective functions, using the ε-constraint method and hence epsilon values. The 10% increase mentioned here is applied for only period 5. In the fifth period, the capacity of machine 1, whose capacity value is 35,400, is analyzed as 38,940 by considering a 10% increase in its capacity. Table 18 shows the status numbers corresponding to the epsilon values for the analysis of change in machine 1 capacity in period 5. The analysis shows that the change in the machine capacity first increases and then decreases the amount of carbon emissions.

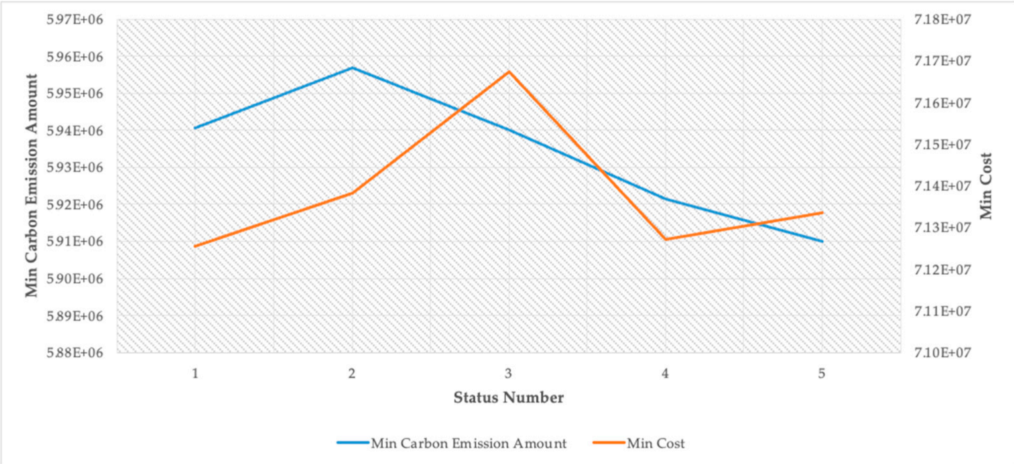


Figure 5. Machine capacity value change for the ε-constraint method.

Table 18. Status numbers regarding epsilon values using the ε-constraint method for the analysis of change in machine 1 capacity in period 5.

Status Number	1	2	3	4	5
Epsilon Value	5,977,181	5,958,612	5,940,043	5,921,474	5,902,905

Objective function values of the model may be affected by alterations in carbon limit values. For instance, Figure 6 shows the impact of a 10% increase in the carbon limit value for cell 6 in the fifth period on both the amount of carbon emission and the total cost. The analysis examines the influence of increasing the carbon limit value on emission amount and total cost using the ε-constraint method and hence epsilon values. In the fifth period of cell number 6, the carbon emission limit value of 357,800 is investigated as 393,580 due to a 10% increment. Table 19 indicates the status numbers regarding the epsilon values for the analysis of change in the carbon limit value for cell 6 in the period 5. As can be seen in the figure, with the increase in carbon emission limit value, the objective functions related to the cost and carbon emission amounts initially show an increase. Then, it is seen that the cost and carbon emission amounts decrease.

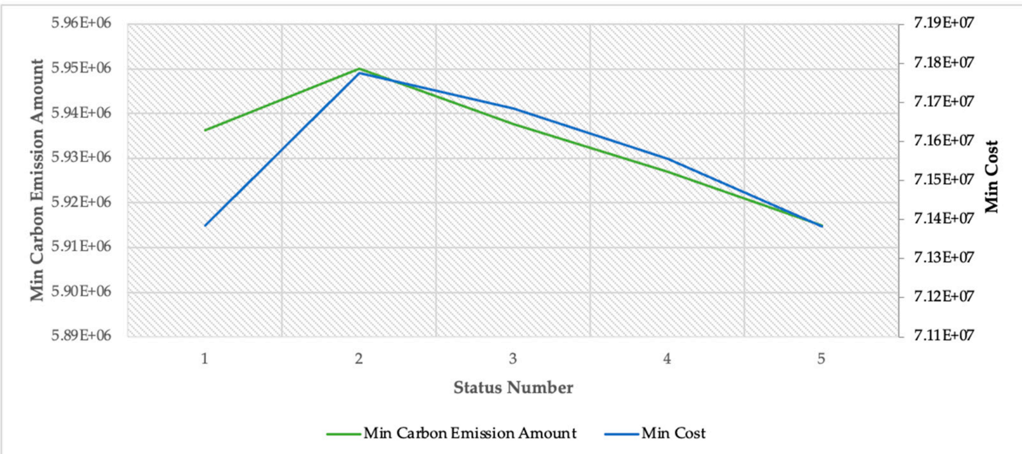


Figure 6. Carbon limit value change for the ε-constraint method.

Table 19. Status number regarding the epsilon value using the ε-constraint method for the analysis of change in the carbon limit value for cell 6 in period 5.

Status Number	**1**	**2**	**3**	**4**	**5**
Epsilon Value	5,963,143	5,951,103	5,939,063	5,927,023	5,914,983

Figure 7 shows the effect on objective functions by changing the maximum number of workers that can be assigned to cells in each period. In the current situation, the maximum number of workers that can be assigned to each cell in each period is taken as four. For this analysis the maximum number of workers for six cells are assumed as 6, 5, 5, 5, 6, and 6 in the first period, 5, 6, 5, 5, 6, and 4 in the second period, 5, 5, 6, 5, 6, and 5 in the third period, 6, 6, 5, 6, 5, and 6 in the fourth period, and 5, 5, 4, 6, 6, and 6 in the fifth period, respectively. The status numbers corresponding to the epsilon values for the analysis of change in the maximum number of workers that can be assigned to cells in each period are shown in Table 20. As seen in Figure 7, the analysis shows that the change in the maximum number of workers leads to a gradual decrease in the amount of carbon emissions.

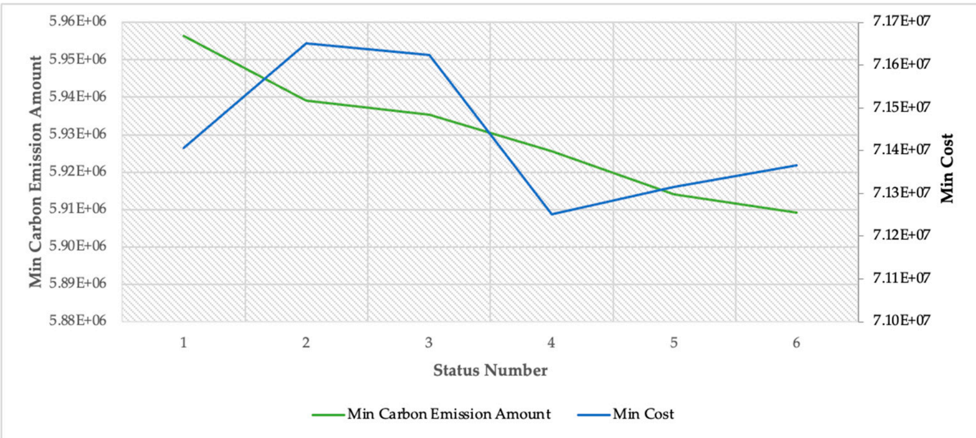


Figure 7. Maximum number of workers in the cell change process for the ε-constraint method.

Table 20. Status number regarding the epsilon value using the ε-constraint method for the analysis of change in the maximum number of workers.

Status Number	**1**	**2**	**3**	**4**	**5**	**6**
Epsilon Value	5,959,367	5,948,075	5,936,783	5,925,491	5,914,199	5,909,193

Alterations in carbon limit values can affect the objective function values of the model. For example, Figure 8 illustrates a 10% increase in carbon limit value for cell 6 in period 5 using the AUGMECON method. Table 21 shows the status numbers regarding the epsilon values using the AUGMECON method for the analysis of change in carbon limit value for cell 6 in the fifth period. As seen in Figure 8, the analysis shows that the change in carbon limit value for cell 6 in period 5 causes a decrease in the amount of carbon emissions.

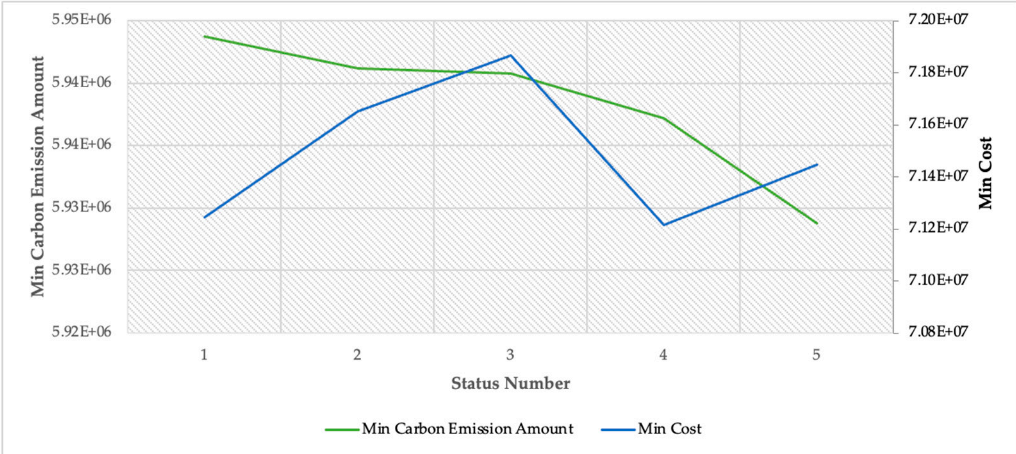


Figure 8. Carbon limit value change for the AUGMECON method.

Table 21. Status number regarding the epsilon value using the AUGMECON method for the analysis of change in the carbon limit value for cell 6 in period 5.

Status Number	1	2	3	4	5
Epsilon Value	5,962,379	5,953,925	5,945,771	5,937,417	5,929,064

The objective functions of the model can be affected by changes in machine capacity values. For instance, Figure 9 displays the impact of a 10% increase in the capacity values of machine 3 in period 3. The status numbers regarding the epsilon values using the AUGMECON method for the analysis of change in machine 3 capacity in period 3 are indicated in Table 22. As seen in Figure 9, in the analysis, as the machine capacity changes, the carbon emission amount value, which is the objective function, first increases and then decreases.

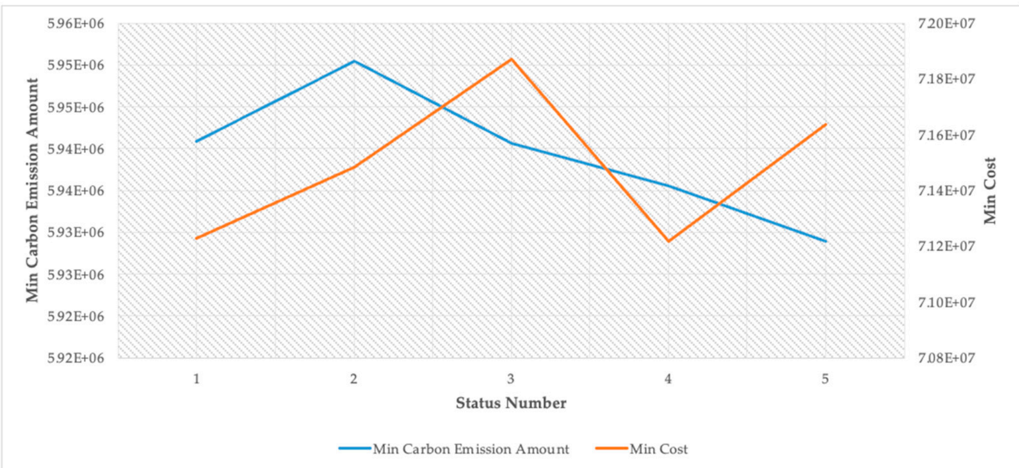


Figure 9. Machine capacity value change for the AUGMECON method.

Table 22. Status number regarding epsilon value using the AUGMECON method for the analysis of change in machine 3 capacity in period 3.

Status Number	1	2	3	4	5
Epsilon Value	5,965,359	5,955,535	5,945,711	5,935,887	5,929,064

5. Conclusions and Future Studies

In this study, three multi-objective mathematical programming models were presented that focus on the design of flexible manufacturing cells while incorporating sustainable factors. The study considers economic, environmental, and social dimensions, which are the three key dimensions of sustainability, by including various parameters. By considering these dimensions, this study aimed to develop the design of flexible manufacturing cells within a sustainable framework. In addition to minimizing the number of carbon emissions within the scope of the environmental dimension, this study aimed to minimize various cost items considering carbon emissions, inter-cellular movement, machine replacement, machine operation, worker training, and bonus wages for workers as the economic dimension. The total training time received by each worker in all periods is shown as a constraint in the model within the scope of the social dimension. Since the study involves multi-objectives, the identified problem is modeled using multi-objective optimization techniques. Firstly, the goal programming model related to the problem was developed. Then, the ε-constraint and AUGMECON models for the examined problem were presented. In all developed multi-objective models, various decision variables were considered to optimize the flexible manufacturing cells. They cover the decision variables such as determining the optimal routes between the alternative routes of parts, the number of machines to be added to or removed from cells, the number of workers assigned to cells, and the total training time of workers. In this study, all the developed multi-objective mathematical programming models were solved using the LINGO 20.0 optimization program on the developed sample problem. These global optimal solutions were reached in 12 min and 38 s, 18 min and 31 s, and 19 min and 51 s for the goal programming method, the ε-constraint method, and the AUGMECON method, respectively. When the results obtained from each of the developed multi-objective optimization models were examined, it was observed that the decision variables regarding determining optimal routes of parts, assigning optimal machines to cells, adding them to cells, and removing them from cells provide different results. While the ε-constraint and AUGMECON models provided the same results in the optimal worker assignments and the optimal number of workers in cells for each period, the goal programming model provided different results. The decision variables of the total number of workers in the system in each period and the total training times received by workers provided the same results for all developed models. The results were tested by performing sensitivity analyzes for each developed multi-objective optimization model.

In future studies, metaheuristic algorithms can be proposed to solve larger-scale problems in the context of sustainable manufacturing systems. Additionally, the consideration of parameters such as machining times and demands such as fuzzy variables can enhance modeling capabilities and address uncertainties in real-world scenarios. Furthermore, the development of a decision support system specifically designed for modeling sustainable manufacturing systems holds the potential to yield valuable insights for making informed decisions.

Author Contributions: Conceptualization, E.B. and E.Y.; methodology, E.B. and E.Y.; software, E.B.; validation, E.B.; visualization, E.B. and E.Y.; writing—original draft preparation, E.B. and E.Y.; and writing—review and editing, E.B. and E.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in this article.

Acknowledgments: Thanks to LINDO Systems Inc. for ensuring a free educational license of LINGO 20.0 software package.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Alternative routes of parts, machine process time, and power consumption.

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
1	1	Q1-S1(9)/PW(14) Q2-S4(4)/PW(15) Q3-S7(6)/PW(13) Q4-S11(3)/PW(14)-S12(7)/PW(17) Q5-S14(7)/PW(18)-S15(5)/PW(14) Q6-S20(8)/PW(18)
	2	Q1-S2(6)/PW(13)-S3(5)/PW(15) Q2-S5(7)/PW(17) Q4-S11(7)/PW(19)-S12(8)/PW(17)-S13(4)/PW(15) Q5-S14(9)/PW(18)-S15(4)/PW(14) Q6-S18(5)/PW(14) Q6-S20(7)/PW(18)
	3	Q1-S1(6)/PW(16)-S2(9)/PW(19)-S3(4)/PW(14) Q3-S8(2)/PW(22) Q4-S12(9)/PW(19)-S13(6)/PW(26) Q5-S14(8)/PW(18)-S15(4)/PW(24) Q6-S18(9)/PW(19)-S20(8)/PW(18)
	4	Q1-S1(11)/PW(19)-S2(6)/PW(18)-S3(5)PW(25) Q2-S5(5)/PW(25) Q4-S11(8)/PW(19)-S12(5)/PW(24)-S13(5)/PW(15) Q5-S14(5)/PW(23) Q5-S15(4)/PW(24) Q6-S20(8)/PW(18)
	5	Q1-S1(8)/PW(16)-S2(5)/PW(19)-S3(4)/PW(24) Q3-S8(8)/PW(16) Q4-S12(9)/PW(19)-S13(6)/PW(16) Q5-S14(7)/PW(18)-S15(4)/PW(24) Q6-S18(8)/PW(19)-S20(5)/PW(18)
2	1	Q1-S3(8)/PW(18) Q2-S6(5)/PW(15) Q3-S7(4)/PW(14) Q4-S9(8)PW(14)-S10(6)/PW(15) Q5-S13(7)/PW(18)-S14(9)/PW(19) Q6-S19(4)/PW(24)
	2	Q1-S1(6)/PW(20) Q2-S6(7)/PW(18) Q4-S11(8)/PW(18)-S13(5)/PW(23) Q5-S14(6)/PW(27) Q6-S18(8)/PW(18)-S19(3)/PW(22)
	3	Q1-S2(5)/PW(25)-S3(3)/PW(23) Q2-S6(6)/PW(16) Q3-S7(4)/PW(24)-S8(7)/PW(17) Q4-S8(8)/PW(8) Q5-S16(4)/PW(14) Q6-S20(7)/PW(17)
	4	Q1-S1(8)/PW(18)-S2(9)/PW(17)-S3(3)/PW(23) Q2-S6(8)/PW(8) Q4-S11(8)/PW(8)-S13(3)/PW(23) Q5-S14(7)/PW(17) Q6-S18(8)/PW(18)-S19(3)/PW(22)
	5	Q1-S2(5)/PW(15)-S3(3)/PW(23) Q3-S7(5)/PW(24)-S8(7)/PW(17) Q4-S8(7)/PW(18) Q5-S15(5)/PW(25)-S16(4)/PW(24) Q6-S20(9)/PW(17)
3	1	Q1-S1(9)/PW(12)-S3(4)/PW(14) Q2-S6(6)/PW(17) Q4-S10(5)/PW(17)-S12(4)/PW(22) Q5-S15(5)/PW(18) Q6-S19(7)/PW(14)-S20(6)/PW(15)
	2	Q1-S1(2)/PW(23) Q2-S4(5)/PW(15)-S5(7)/PW(18) Q3-S8(6)/PW(26) Q5-S16(4)/PW(24) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(7)/PW(17)-S2(5)/PW(21)-S3(3)/PW(13) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(5)/PW(23)-S2(7)/PW(17) Q2-S4(1)/PW(19)-S5(6)/PW(18) Q3-S8(5)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(27)
	5	Q1-S1(8)/PW(11)-S2(1)/PW(17) Q2-S5(8)/PW(18) Q3-S8(2)/PW(14)-S9(6)/PW(16) Q6-S20(7)/PW(17)
4	1	Q1-S1(5)/PW(21)-S2(7)/PW(22)-S3(3)/PW(14) Q2-S4(5)/PW(15)-S6(8)/PW(17) Q4-S10(7)/PW(17)-S12(4)/PW(12) Q5-S15(6)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(6)/PW(17)-S2(9)/PW(22)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(23)-S2(4)/PW(17) Q2-S4(6)/PW(21)-S5(1)7PW(18) Q3-S8(5)/PW(16)-S9(6)/PW(26) Q6-S20(7)/PW(17)
	5	Q1-S1(4)/PW(21)-S2(6)/PW(17) Q2-S5(7)/PW(18) Q3-S8(8)/PW(24)-S9(6)/PW(26) Q6-S20(3)/PW(27)
5	1	Q1-S1(2)/PW(14)-S2(5)/PW(15)-S3(4)/PW(19) Q2-S6(8)/PW(18)-S7(10)/PW(19) Q4-S10(5)/PW(15)-S12(2)/PW(12) Q5-S15(6)/PW(16)-S16(5)/PW(15) Q6-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(11)/PW(17) Q2-S4(9)/PW(19)-S5(8)/PW(18)-S6(6)/PW(16) Q3-S8(6)/PW(16)-S9(7)/PW(7) Q5-S16(4)/PW(24)-S17(5)/PW(25) Q6-S18(5)/PW(15)-S19(3)/PW(23)-S20(4)/PW(14)
	3	Q1-S1(8)/PW(18)-S3(3)/PW(23) Q2-S5(9)/PW(19) Q3-S7(7)/PW(17) Q4-S15(9)/PW(19)-S16(8)/PW(18)
	4	Q1-S1(6)/PW(16)-S2(8)/PW(18)-S3(5)/PW(25) Q2-S4(4)/PW(24)-S5(7)/PW(17)-S6(6)/PW(26) Q3-S8(6)/PW(16)-S9(5)/PW(25) Q6-S19(4)/PW(24)-S20(7)/PW(17)
	5	Q1-S1(5)/PW(15)-S2(6)/PW(16) Q2-S5(8)/PW(18) Q3-S8(8)/PW(18)-S9(6)/PW(16) Q6-S19(6)/PW(19)
6	1	Q1-S1(9)/PW(17)-S2(6)/PW(15)-S3(8)/PW(18) Q2-S6(7)/PW(17)-S7(6)/PW(26) Q4-S10(7)/PW(17)-S12(2)/PW(22) Q5-S15(8)/PW(28) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(9)/PW(15)-S5(8)/PW(8) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(8)/PW(17)-S2(2)/PW(22)-S3(1)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(5)/PW(13)-S2(7)/PW(17) Q2-S4(2)/PW(19)-S5(4)/PW(18) Q3-S8(5)/PW(16)-S9(6)/PW(16) Q6-S20(3)/PW(17)
	5	Q1-S1(6)/PW(18)-S2(3)/PW(17) Q2-S5(8)/PW(8) Q3-S8(2)/PW(4)-S9(6)/PW(6) Q6-S20(6)/PW(7)
7	1	Q1-S3(8)/PW(24) Q2-S5(5)/PW(8)-S6(7)/PW(17) Q4-S10(7)/PW(17)-S12(8)/PW(12) Q5-S15(8)/PW(8) Q6-S19(4)/PW(4)-S20(5)/PW(5)
	2	Q1-S1(3)/PW(23) Q2-S4(7)/PW(5)-S5(8)/PW(8) Q3-S8(6)/PW(6) Q5-S16(5)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(4)
	3	Q1-S1(7)/PW(7)-S2(1)/PW(18)-S3(3)/PW(13) Q2-S5(5)/PW(5) Q3-S7(6)/PW(6) Q4-S15(6)/PW(7)
	4	Q1-S1(4)/PW(3)-S2(7)/PW(7) Q2-S4(3)/PW(11)-S5(8)/PW(8) Q3-S8(9)/PW(6)-S9(6)/PW(6) Q6-S20(7)/PW(7)
	5	Q1-S1(5)/PW(12)-S2(6)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(5)/PW(16) Q6-S20(6)/PW(7)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
8	1	Q1-S1(9)/PW(9)-S3(8)/PW(8) Q2-S6(8)/PW(7) Q4-S10(5)/PW(7)-S12(3)/PW(12) Q5-S15(6)/PW(8) Q6-S19(5)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(9)/PW(9) Q2-S4(6)/PW(16)-S5(8)/PW(8) Q3-S8(6)/PW(6) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(7)/PW(17)-S2(5)/PW(15)-S3(3)/PW(13) Q2-S5(5)/PW(5) Q3-S7(6)/PW(16) Q4-S15(7)/PW(7)
	4	Q1-S1(7)PW(13)-S2(7)/PW(7) Q2-S4(9)/PW(11)-S5(8)/PW(8) Q3-S8(10)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(7)
	5	Q1-S1(4)/PW(14)-S2(7)/PW(7) Q2-S5(8)/PW(8) Q3-S8(5)/PW(14)-S9(6)/PW(16) Q6-S20(2)/PW(7)
9	1	Q1-S3(8)/PW(8) Q2-S5(8)/PW(18)-S6(3)/PW(17)-S7(12)/PW(19) Q4-S11(7)/PW(9)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S19(4)/PW(4)-S20(5)/PW(5)
	2	Q1-S1(5)/PW(8)-S2(9/PW(9) Q2-S4(6)/PW(15)-S5(8)/PW(8) Q3-S7(7)/PW(7)-S8(6)/PW(6) Q5-S16(4)/PW(14)-S17(6)/PW(6) Q6-S20(4)/PW(4)
	3	Q1-S1(9)/PW(7)-S3(7)/PW(13) Q2-S5(6)/PW(5)-S6(8)/PW(8) Q3-S7(6)/PW(6)-S8(8)/PW(8) Q4-S15(7)/PW(17)-S16(9)/PW(9)
	4	Q1-S1(3)/PW(13)-S2(7)/PW(7)-S3(4)/PW(14) Q2-S4(6)/PW(11)-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(6) Q6-S20(2)/PW(7)
	5	Q1-S1(3)/PW(15)-S2(5)/PW(7)-S3(7)/PW(7) Q2-S5(8)/PW(8) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q6-S19(3)/PW(18)-S20(6)/PW(17)
10	1	Q2-S5(4)/PW(14)-S6(4)/PW(15) Q4-S10(7)/PW(17)-S11(8)/PW(18)-S12(5)/PW(22) Q5-S15(8)/PW(8)-S17(9)/PW(9) Q6-S19(1)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(5)/PW(25) Q2-S4(5)/PW(25)-S5(8)/PW(28)-S6(4)/PW(24) Q3-S8(6)/PW(6) Q5-S16(4)/PW(14)-S17(5)/PW(15) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(9)/PW(16)-S2(8)/PW(18)-S3(3)/PW(13) Q2-S5(5)/PW(15)-S6(6)/PW(17) Q3-S7(6)/PW(16)-S8(8)/PW(18) Q4-S15(7)/PW(17)
	4	Q1-S1(5)/PW(15)-S2(7)/PW(17) Q2-S4(4)/PW(14)-S5(8)/PW(8) Q3-S8(6)/PW(16)-S9(6)/PW(16) Q6-S18(8)/PW(18)-S20(7)/PW(17)
	5	Q1-S1(3)/PW(14)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(26) Q6-S19(4)/PW(17)
11	1	Q1-S2(8)/PW(18) Q2-S5(4)/PW(14)-S6(7)7PW(17) Q4-S10(4)/PW(14)-S11(8)/PW(18)-S12(5)/PW(15) Q5-S15(8)/PW(8)-S16(7)/PW(7) Q6-S18(6)/PW(16)-S19(6)/PW(18)-S20(9)/PW(9)
	2	Q1-S1(8)/PW(18)-S2(6)/PW(8) Q2-S4(5)/PW(15)-S5(4)/PW(8)-S6(9)/PW(9) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(11)-S20(4)/PW(14)
	3	Q1-S2(5)/PW(15) Q2-S6(7)/PW(17)-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(10)/PW(19)-S2(7)/PW(17)-S3(9)/PW(9) Q2-S4(3)/PW(8)-S5(8)/PW(8) Q3-S8(5)/PW(15)-S9(6)/PW(16) Q6-S18(4)/PW(14)-S19(8)/PW(18)-S20(7)/PW(17)
	5	Q1-S1(9)/PW(16)-S2(7)/PW(17) Q2-S4(7)/PW(17)-S5(8)/PW(8) Q3-S7(5)/PW(15)-S8(4)/PW(24)-S9(6)/PW(16) Q6-S19(4)/PW(18)-S20(6)/PW(16)
12	1	Q1-S3(8)/PW(13) Q2-S6(6)/PW(16)-S7(5)/PW(15) Q4-S10(4)/PW(14)-S11(3)/PW(13)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S20(5)/PW(15)
	2	Q1-S1(3)/PW(13)-S2(7)/PW(17) Q2-S4(7)/PW(26)-S5(7)/PW(17) Q3-S8(7)/PW(14) Q5-S16(5)/PW(15)-S17(4)/PW(14) Q6-S19(7)/PW(17)-S20(8)/PW(18)
	3	Q1-S1(2)/PW(22)-S2(8)/PW(18) Q2-S5(7)/PW(17)-S6(4)/PW(24) Q3-S7(6)/PW(6)-S8(5)/PW(5) Q4-S15(7)/PW(19)-S16(8)/PW(18)
	4	Q1-S1(6)/PW(7)-S2(6)/PW(16) Q2-S4(5)/PW(15)-S5(4)/PW(24) Q3-S8(4)/PW(24) Q6-S20(7)/PW(17)
	5	Q1-S2(1)/PW(8) Q2-S5(11)/PW(18) Q3-S8(2)/PW(24)-S9(6)/PW(16) Q6-S19(8)/PW(18)
13	1	Q1-S2(7)/PW(18)-S3(5)/PW(12) Q2-S4(3)/PW(16)-S6(7)/PW(17) Q4-S10(6)/PW(17)-S11(3)/PW(18)-S12(2)/PW(22) Q5-S15(8)/PW(18)-S16(9)/PW(9) Q6-S18(7)/PW(17)-S19(6)/PW(16)-S20(5)/PW(25)
	2	Q1-S1(5)/PW(15)-S2(7)/PW(26) Q2-S4(5)/PW(15)-S5(4)/PW(18)-S6(7)/PW(17) Q3-S8(6)/PW(6)-S9(7)/PW(7) Q5-S16(4)/PW(24)-S17(5)/PW(15) Q6-S19(6)/PW(16)
	3	Q1-S3(3)/PW(13) Q2-S5(3)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S15(7)/PW(17)-S16(6)/PW(16)
	4	Q1-S2(10)/PW(17) Q2-S5(8)/PW(18) Q3-S8(9)/PW(15)-S9(6)/PW(16)-S10(4)/PW(24) Q6-S19(8)/PW(18)
	5	Q1-S1(6)/PW(24)-S2(7)/PW(17) Q2-S5(1)/PW(21) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q6-S18(9)/PW(9)-S19(8)/PW(18)
14	1	Q2-S6(9)/PW(17) Q4-S10(5)/PW(21)-S12(2)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(24) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(8)/PW(17)-S2(1)/PW(22)-S3(3)/PW(13) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(4)/PW(13)-S2(7)(PW(17) Q2-S4(1)/PW(21)-S5(5)/PW(28) Q3-S8(6)/PW(26)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S1(11)/PW(21)-S2(3)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q6-S20(4)/PW(17)
15	1	Q1-S1(11)/PW(18) Q2-S6(7)/PW(17)-S7(5)/PW(15) Q3-S15(8)/PW(8) Q4-S10(3)/PW(17)-S11(5)/PW(15)-S12(2)/PW(23) Q5-S15(8)/PW(8) Q6-S18(9)/PW(19)-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(10)/PW(19)-S3(3)/PW(23) Q2-S5(5)/PW(25) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S4(9)/PW(19)-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(3)/PW(16)-S10(7)/PW(17) Q6-S19(6)/PW(16) S20(9)/PW(19)
	5	Q1-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S9(6)/PW(16) Q6-S19(3)/PW(18)-S20(7)/PW(17)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
16	1	Q4-S10(7)/PW(17)-S12(5)/PW(20) Q5-S15(8)/PW(18)
	2	Q2-S4(12)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(2)/PW(24) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(4)/PW(17)-S2(8)/PW(22)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(13)-S2(7)/PW(17) Q2-S4(6)/PW(18)-S5(3)/PW(18) Q3-S8(7)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S1(7)/PW(17)-S2(3)/PW(19)-S3(5)/PW(15) Q2-S5(8)/PW(18) Q3-S6(4)/PW(14)-S8(4)/PW(14)-S9(6)/PW(16) Q6-S19(5)/PW(15)-S20(2)/PW(17)
17	1	Q1-S2(6)/PW(17)-S3(5)/PW(15) Q2-S6(7)/PW(17) Q4-S10(1)/PW(17)-S12(2)/PW(22) Q5-S15(6)/PW(18)-S16(5)/PW(25) Q6-S18(6)/PW(26)-S20(5)/PW(25)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S4(5)/PW(15) Q3-S7(4)/PW(14)-S8(6)/PW(16) Q5-S16(11)/PW(24)-S17(7)/PW(17) Q6-S20(9)/PW(19)
	3	Q1-S1(8)/PW(19)-S3(7)/PW(14) Q2-S5(5)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(26)-S8(5)/PW(15) Q4-S15(7)/PW(17)-S16(8)/PW(18) Q5-S18(9)/PW(19)
	4	Q1-S1(6)/PW(23)-S2(7)/PW(27) Q2-S4(1)/PW(21)-S5(8)/PW(18) Q3-S8(4)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S2(4)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16)
18	1	Q1-S2(9)/PW(19)-S3(4)/PW(24) Q2-S6(7)/PW(17) Q4-S10(7)/PW(27)-S12(9)/PW(19) Q5-S15(8)/PW(18)-S19(4)/PW(24)-S20(5)/PW(25) Q6-S18(5)PW(15)
	2	Q1-S1(9)/PW(19)-S2(5)/PW(25) Q2-S4(4)/PW(24) Q3-S8(6)/PW(16)-S9(6)/PW(16) Q5-S16(6)/PW(16)-S17(7)/PW(17) Q6-S19(8)/PW(18)-S20(4)/PW(24)
	3	Q1-S1(9)/PW(19)-S3(7)/PW(17) Q2-S5(5)PW(15) Q3-S7(6)/PW(16)-S8(5)/PW(15) Q4-S15(9)/PW(19)-S16(4)/PW(24)
	4	Q1-S2(9)/PW(9)-S3(4)/PW(14) Q2-S5(3)/PW(13)-S6(5)/PW(15) Q3-S8(6)/PW(16) Q6-S18(3)/PW(13)-S19(8)/PW(8)-S20(4)/PW(14)
	5	Q1-S1(6)/PW(16)-S2(7)/PW(17)-S3(5)/PW(25) Q2-S5(8)/PW(8)-S8(4)/PW(14) Q3-S8(4)/PW(14)-S9(6)/PW(16)-S10(5)/PW(15) Q6-S19(5)/PW(15)-S20(7)/PW(17)
19	1	Q2-S6(4)/PW(14)-S7(8)/PW(18) Q4-S10(7)/PW(17)-S11(5)/PW(25)-S12(2)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(6)/PW(16) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(26) Q5-S16(6)/PW(26) Q6-S19(13)/PW(13)-S20(4)/PW(14)
	3	Q1-S2(9)/PW(9)-S3(8)/PW(18) Q2-S5(5)/PW(15) Q3-S7(11)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S2(9)/PW(19) Q2-S4(7)/PW(11)-S5(8)/PW(18) Q3-S8(7)/PW(7)-S9(6)/PW(16) Q6-S19(2)/PW(22)-S20(3)/PW(3)
	5	Q2-S5(8)/PW(8) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q6-S19(9)/PW(15)-S20(6)/PW(16)
20	1	Q2-S6(11)/PW(9)-S7(9)/PW(9) Q3-S9(5)PW(15) Q4-S10(6)/PW(16)-S11(8)/PW(18)-S12(2)/PW(22) Q5-S15(9)/PW(9) Q6-S18(6)/PW(16)-S19(4)/PW(24)-S20(5)/PW(25)
	2	Q1-S1(3)/PW(13)-S2(8)/PW(18) Q2-S4(9)/PW(9)-S5(6)/PW(16) Q3-S7(7)/PW(17)-S8(6)/PW(16) Q6-S20(4)/PW(14)
	3	Q1-S2(12)/PW(12) Q2-S5(9)/PW(9) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S15(8)/PW(8)-S16(9)/PW(9) Q6-S19(8)/PW(8)
	4	Q1-S2(7)/PW(17)-S3(8)/PW(18) Q3-S8(3)/PW(16)-S9(6)/PW(16)-S10(5)/PW(25) Q5-S17(9)/PW(9) Q6-S19(5)/PW(25)-S20(8)/PW(18)
	5	Q2-S5(8)/PW(8) Q3-S8(7)/PW(9)-S9(10)/PW(6)
21	1	Q1-S1(9)/PW(8)-S2(5)/PW(9)-S3(4)/PW(14) Q4-S10(8)/PW(15)-S11(6)/PW(16)-S12(2)/PW(22) Q6-S19(4)/PW(24)-S20(5)/PW(25)
	2	Q2-S4(9)/PW(15)-S5(8)/PW(18) Q3-S8(8)/PW(16)-S9(7)/PW(17) Q6-S19(9)/PW(9)
	3	Q1-S3(9)/PW(9) Q2-S5(5)/PW(25)-S6(7)/PW(17) Q3-S7(6)/PW(26) Q4-S15(7)/PW(17) Q6-S18(8)/PW(18)-S19(9)/PW(19)
	4	Q1-S2(7)/PW(17) Q2-S4(13)/PW(11)-S5(4)/PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S1(4)/PW(24)-S2(6)/PW(16) Q2-S5(5)/PW(17) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q4-S11(7)/PW(17)
22	1	Q1-S1(5)/PW(16)-S3(2)/PW(24) Q2-S5(5)/PW(18) Q4-S10(7)/PW(17)-S11(9)/PW(9)-S12(7)/PW(17) Q5-S15(8)/PW(8)-S16(5)/PW(15) Q6-S20(9)/PW(9)
	2	Q1-S1(5)/PW(3)-S2(9)/PW(9) Q3-S8(6)/PW(16)-S9(8)/PW(8) Q5-S16(4)/PW(14)-S17(5)/PW(15) Q6-S19(9)/PW(9)
	3	Q1-S1(8)/PW(18)-S2(2)/PW(12) Q2-S5(9)/PW(9) Q3-S7(7)/PW(16)-S8(7)/PW(17) Q4-S15(7)/PW(17)-S16(8)/PW(18)
	4	Q2-S5(8)/PW(8)-S6(7)/PW(7) Q3-S8(6)/PW(6)-S9(8)/PW(9) Q6-S18(3)/PW(3)-S20(9)/PW(9)
	5	Q1-S1(7)/PW(17)-S2(9)/PW(9) Q2-S4(2)/PW(20)-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16)-S10(3)/PW(23) Q4-S16(4)/PW(14) Q5-S17(6)/PW(16)-S18(5)/PW(15) Q6-S20(7)/PW(17)
23	1	Q1-S3(3)/PW(14) Q2-S6(7)/PW(14) Q5-S15(8)/PW(18)-S16(9)/PW(9)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S4(5)/PW(15)-S5(6)/PW(18)-S6(7)/PW(17) Q3-S8(6)/PW(16) Q4-S15(6)/PW(16) Q6-S20(9)/PW(9)
	3	Q1-S1(9)/PW(9)-S2(8)/PW(8) Q2-S4(4)/PW(24)-S5(5)/PW(25) Q3-S7(6)/PW(26)-S8(7)/PW(17)
	4	Q2-S4(2)/PW(21)-S5(5)/PW(17)-S6(8)/PW(18) Q6-S19(3)/PW(23)-S20(9)/PW(9)
	5	Q2-S5(6)/PW(16) Q3-S9(6)/PW(6) Q6-S18(11)/PW(9)-S20(3)/PW(15)
24	1	Q1-S1(9)/PW(8)-S2(2)/PW(13)-S3(1)/PW(14) Q2-S5(6)/PW(16)-S6(7)/PW(17) Q4-S10(6)/PW(14)-S11(8)/PW(18) Q5-S14(6)/PW(16)-S15(8)/PW(18) Q6-S19(6)/PW(16)-S20(3)/PW(13)
	2	Q2-S5(7)/PW(16)-S6(7)/PW(17) Q3-S8(9)/PW(19)-S9(5)/PW(15) Q5-S16(3)/PW(23)-S17(2)/PW(22) Q6-S20(4)/PW(12)
	3	Q1-S1(6)/PW(16)-S3(8)/PW(9) Q4-S15(4)/PW(4)-S16(5)/PW(5) Q6-S19(8)/PW(8)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17)-S3(6)/PW(16) Q2-S5(8)/PW(8)-S6(5)/PW(15) Q6-S19(8)/PW(18)-S20(9)/PW(9)
	5	Q1-S2(7)/PW(17) Q2-S5(8)/PW(18)-S6(5)/PW(15) Q3-S8(4)/PW(14)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
25	1	Q1-S1(5)/PW(15)-S2(6)/PW(16)-S3(5)/PW(14) Q2-S5(8)PW(18)-S6(7)/PW(17)-S7(5)/PW(15) Q3-S8(5)/PW(23) Q4-S12(9)/PW(9) Q5-S15(8)/PW(18)
	2	Q1-S1(3)/PW(3)-S2(1)/PW(5) Q2-S4(4)/PW(24)-S5(7)/PW(17) Q3-S8(5)/PW(15)-S9(4)PW(24) Q5-S16(3)/PW(8) Q6-S19(9)/PW(9)-S20(6)/PW(6)
	3	Q1-S2(8)/PW(18)-S3(9)/PW(9) Q2-S5(5)/PW(15) Q3-S6(4)/PW(10)-S7(6)/PW(16)
	4	Q1-S1(6)/PW(16)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S4(2)/PW(12)-S5(8)/PW(18)-S6(3)/PW(11) Q3-S8(6)/PW(16)-S9(6)/PW(16) Q6-S19(8)/PW(18)-S20(7)/PW(17)
	5	Q1-S2(9)/PW(17) Q2-S4(4)/PW(14)-S5(8)/PW(8) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q6-S19(9)/PW(16)-S20(7)/PW(17)
26	1	Q1-S3(4)/PW(14) Q2-S6(10)/PW(17)-S7(8)/PW(18) Q4-S10(7)/PW(17) Q5-S15(8)/PW(8)-S16(8)/PW(8) Q6-S18(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(7)/PW(17)-S2(8)/PW(18) Q2-S4(2)/PW(21)-S5(3)/PW(13)-S6(4)/PW(24) Q3-S7(5)/PW(15)-S8(6)/PW(16)-S9(7)/PW(17) Q5-S16(4)/PW(14)-S17(5)/PW(15) Q6-S18(7)/PW(17)S20(4)/PW(24)
	3	Q1-S1(10)/PW(18)-S2(9)/PW(19) Q2-S5(5)/PW(15)-S6(6)/PW(16) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S14(7)/PW(17)-S15(8)/PW(18)
	4	Q2-S4(8)/PW(18)-S5(8)/PW(18) Q3-S8(9)/PW(19)
	5	Q1-S2(4)/PW(14) Q2-S5(6)/PW(16)-S6(7)/PW(17) Q3-S8(5)/PW(15)-S9(7)/PW(17) Q5-S16(6)/PW(16)-S17(3)/PW(23) Q6-S18(2)/PW(14)-S19(8)/PW(18)
27	1	Q1-S1(7)/PW(9)-S2(1)/PW(21)-S3(4)/PW(14) Q2-S4(3)/PW(23)-S6(5)/PW(15) Q4-S10(5)/PW(15)-S11(6)/PW(16)-S12(2)/PW(12) Q6-S19(7)/PW(17)
	2	Q1-S1(15)/PW(13)-S2(5)/PW(15)-S3(1)/PW(16) Q2-S5(5)/PW(15)-S6(2)/PW(2) Q3-S7(3)/PW(17)-S8(6)/PW(16) Q5-S6(4)/PW(4)-S17(5)PW(5)
	3	Q1-S1(6)/PW(16)-S3(3)/PW(23) Q2-S5(8)/PW(18)-S6(9)/PW(9) Q3-S7(7)/PW(17)-S8(5)/PW(15) Q4-S14(6)/PW(26)-S15(7)/PW(7) Q6-S19(8)/PW(18)-S20(4)/PW(14)
	4	Q1-S1(3)/PW(3)-S2(8)/PW(6)-S3(4)/PW(4) Q4-S5(7)/PW(8) Q6-S20(7)/PW(7)
	5	Q1-S1(4)/PW(24)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S4(5)/PW(25)-S5(8)/PW(18) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q5-S17(6)/PW(16) Q6-S19(5)/PW(17)-S20(7)/PW(17)
28	1	Q1-S1(4)/PW(15)-S3(6)/PW(16) Q2-S6(9)/PW(9) Q4-S10(6)/PW(16)-S12(3)/PW(13) Q5-S15(8)/PW(18) Q6-S19(6)/PW(16)-S20(8)/PW(18)
	2	Q1-S1(3)/PW(16) Q2-S4(12)/PW(22)-S5(8)/PW(18) Q3-S8(7)/PW(17) Q5-S16(6)/PW(6) Q6-S19(5)/PW(15)-S20(1)/PW(23)
	3	Q1-S1(9)/PW(19)-S3(6)/PW(26) Q2-S5(5)/PW(5) Q3-S7(7)/PW(7) Q4-S15(8)/PW(8)
	4	Q1-S1(4)/PW(14)-S2(8)/PW(18) Q2-S4(3)/PW(13)-S5(7)/PW(11) Q3-S8(9)PW(19)-S9(6)/PW(26) Q6-S19(4)/PW(14)-S20(8)/PW(18)
	5	Q1-S1(6)/PW(16)-S2(8)/PW(18) Q2-S5(8)(PW(18) Q3-S8(9)/PW(19)-S9(5)/PW(15) Q6-S19(3)/PW(23)-S20(6)/PW(16)
29	1	Q1-S1(3)/PW(15)-S3(3)/PW(13) Q2-S6(8)/PW(8) Q4-S12(9)/PW(9) Q6-S20(6)/PW(16)
	2	Q2-S4(5)/PW(9)-S5(6)/PW(16)-S6(5)/PW(15) Q3-S7(7)/PW(17)-S8(8)/PW(8) Q4-S9(9)/PW(9) Q5-S16(3)/PW(23)-S17(2)/PW(20) Q6-S20(4)/PW(14)
	3	Q1-S2(6)/PW(9)-S3(3)/PW(13) Q3-S7(7)/PW(17)-S8(8)/PW(18) Q6-S18(9)/PW(9)
	4	Q1-S2(7)/PW(17) Q2-S5(8)/PW(18)-S6(4)/PW(14) Q3-S7(5)/PW(15)-S8(4)/PW(16)-S9(6)/PW(16) Q6-S19(5)/PW(15)-S20(6)/PW(16)
	5	Q1-S2(7)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q5-S16(4)/PW(14)-S17(5)PW(15) Q6-S19(3)/PW(13)-S20(6)/PW(9)
30	1	Q1-S1(9)/PW(9)-S2(1)/PW(13)-S3(5)/PW(15) Q2-S5(6)/PW(16)-S7(7)/PW(17) Q3-S8(5)/PW(15)-S9(4)/PW(24) Q5-S15(6)/PW(16) Q6-S20(9)/PW(19)
	2	Q1-S1(9)/PW(13)-S2(5)/PW(15)-S3(8)/PW(18) Q2-S4(7)/PW(14)-S5(3)/PW(13) Q3-S7(9)/PW(13)-S8(9)/PW(9) Q5-S16(6)/PW(6)-S17(7)/PW(7)
	3	Q1-S2(7)/PW(17)-S3(3)PW(13) Q2-S5(6)/PW(16)-S6(7)/PW(17) Q3-S7(11)/PW(11) Q4-S15(3)/PW(23)-S16(4)/PW(14) Q6-S18(5)/PW(15)-S19(6)/PW(16)
	4	Q1-S1(9)/PW(9)-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S8(5)/PW(15)-S9(4)/PW(14) Q6-S19(8)/PW(8)-S20(9)/PW(9)
	5	Q1-S2(9)/PW(9)-S3(7)/PW(17) Q2-S4(6)/PW(16)-S5(5)/PW(15) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q4-S11(6)/PW(16)-S12(8)/PW(8) Q5-S16(9)/PW(9)-S17(8)/PW(8)
31	1	Q1-S1(5)/PW(18)-S3(5)PW(15) Q2-S6(6)/PW(16) Q4-S10(6)/PW(16)-S12(3)/PW(23) Q5-S15(9)/PW(19) Q6-S19(7)-S20(3)
	2	Q1-S1(6)/PW(16) Q2-S4(7)/PW(17)-S5(8)/PW(18) Q3-S8(9)/PW(19) Q5-S16(5)/PW(25) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	3	Q1-S1(6)/PW(16)-S2(3)/PW(13)-S3(5)/PW(15) Q2-S5(9)/PW(19) Q3-S7(7)/PW(17) Q4-S15(9)/PW(19)
	4	Q1-S1(9)/PW(17)-S2(3)/PW(23) Q2-S4(2)/PW(20)-S5(4)/PW(14) Q3-S8(9)/PW(19)-S9(3)/PW(23) Q6-S20(8)/PW(18)
	5	Q1-S1(3)/PW(25)-S2(9)/PW(19) Q2-S5(4)/PW(24) Q3-S8(6)/PW(23)-S9(2)/PW(17) Q6-S20(8)/PW(18)
32	1	Q3-S7(6)/PW(16) Q4-S10(8)/PW(18)-S11(7)/PW(17)-S12(1)/PW(23) Q5-S15(4)/PW(14) Q6-S20(5)/PW(15)
	2	Q1-S1(13)/PW(13)-S2(8)/PW(18)-S3(4)/PW(24) Q2-S5(3)/PW(13) Q3-S7(5)/PW(13)-S8(7)/PW(18) Q5-S16(6)/PW(16)-S17(7)/PW(17) Q6-S19(8)/PW(18)
	3	Q1-S1(9)/PW(9) Q3-S7(6)/PW(16)-S8(9)/PW(9) Q4-S14(6)/PW(16)-S15(7)/PW(17) Q5-S16(9)/PW(9)-S17(3)/PW(13) Q6-S18(8)/PW(8)-S19(9)/PW(9)
	4	Q1-S1(6)/PW(16)-S2(10)/PW(17)-S3(5)/PW(10) Q2-S4(2)/PW(5) Q3-S8(3)/PW(13)-S9(7)/PW(7) Q5-S16(5)/PW(15) Q6-S18(5)/PW(15)-S19(9)/PW(9)
	5	Q1-S2(9)/PW(9) Q3-S8(6)/PW(13)-S9(2)/PW(21) Q6-S19(3)/PW(16)-S20(7)/PW(7)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
33	1	Q1-S1(3)/PW(9)-S2(3)/PW(12)-S3(9)/PW(23) Q2-S5(9)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(16)-S8(5)/PW(20) Q4-S10(7)/PW(17)-S11(3)/PW(13)-S12(2)/PW(21) Q5-S15(8)/PW(18)-S16(4)/PW(18) Q6-S18(6)/PW(16)-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(13)-S2(8)/PW(18) Q2-S4(5)/PW(13)-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(7)/PW(17) Q5-S16(4)/PW(14)-S17(5)/PW(23) Q6-S18(5)/PW(15)-S20(4)/PW(14)
	3	Q1-S1(7)/PW(16)-S3(10)/PW(23) Q2-S5(6)/PW(16) Q3-S7(3)/PW(19) Q4-S15(8)/PW(18)-S16(9)/PW(19)
	4	Q1-S2(9)/PW(19)-S3(5)/PW(15) Q2-S4(3)/PW(11)-S5(3)/PW(9) Q3-S7(4)/PW(14) Q4-S10(5)/PW(15)-S11(3)/PW(16)
	5	Q1-S2(8)/PW(18)-S3(9)/PW(19) Q2-S4(4)/PW(24)-S5(5)/PW(12) Q3-S8(3)PW(13)-S9(6)/PW(16)-S10(2)/PW(14) Q5-S16(2)/PW(20)-S17(3)/PW(13) Q6-S18(4)/PW(15)-S19(3)/PW(16)-S20(7)/PW(17)
34	1	Q4-S10(6)/PW(16)-S11(5)/PW(18)-S12(2)PW(22) Q5-S15(6)/PW(18)-S16(8)/PW(18) Q6-S19(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S5(6)/PW(18)-S6(7)/PW(21) Q3-S8(5)/PW(22)-S9(4)/PW(18)-S10(3)/PW(13) Q5-S16(4)/PW(14)-S17(5) Q6-S18(7)/PW(17)-S20(4)/PW(20)
	3	Q1-S1(8)/PW(9)-S3(5)/PW(15) Q2-S5(5)/PW(15)-S6(6)/PW(16) Q3-S7(6)/PW(16)-S8(8)/PW(18) Q4-S15(7)/PW(17)-S16(8)/PW(18)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17)-S3(4)/PW(24) Q2-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(5)/PW(15) Q6-S19(8)/PW(8)
	5	Q1-S1(6)/PW(16)-S2(3)/PW(23)-S3(5)/PW(15) Q2-S5(8)/PW(18)-S6(4)/PW(24) Q3-S8(9)/PW(14)-S9(6)/PW(15)-S10(3)/PW(23) Q6-S18(6)/PW(14)-S19(5)/PW(15)-S20(7)/PW(17)
35	1	Q2-S6(8)/PW(18)-S7(9)/PW(19) Q4-S10(7)/PW(17)-S12(2)/PW(22) Q5-S15(8)/PW(18)
	2	Q1-S1(3)/PW(23) Q2-S4(6)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(14)
	3	Q1-S1(7)/PW(17)-S2(2)/PW(20)-S3(3)/PW(15) Q2-S5(5)/PW(5) Q3-S7(6)/PW(6) Q4-S15(7)/PW(7)
	4	Q1-S1(10)/PW(23)-S2(7)/PW(17) Q2-S4(3)/PW(21)-S5(9)/PW(18) Q3-S8(6)/PW(6)-S9(6)/PW(6) Q6-S20(7)/PW(17)
	5	Q1-S1(6)/PW(11)-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S8(1)/PW(15)-S9(6)/PW(16) Q6-S20(2)/PW(17)
36	1	Q1-S1(3)/PW(12) Q2-S4(5)/PW(17) Q3-S7(3)/PW(13) Q4-S11(3)/PW(14)-S12(7)/PW(17) Q5-S14(8)/PW(18)-S15(4)/PW(14) Q6-S20(8)/PW(18)
	2	Q1-S2(3)/PW(23)-S3(5)/PW(18) Q2-S5(7)/PW(17) Q4-S11(9)/PW(19)-S12(6)/PW(17)-S13(5)/PW(15) Q5-S14(8)/PW(18)-S15(4)/PW(14) Q6-S18(4)/PW(14)-S20(8)/PW(18)
	3	Q1-S1(6)/PW(16)-S2(9)/PW(19)-S3(1)/PW(14) Q3-S8(2)/PW(22) Q4-S12(9)/PW(19)-S13(6)/PW(26) Q5-S14(8)/PW(18)-S15(4)/PW(24) Q6-S18(9)/PW(19)-S20(8)/PW(18)
	4	Q1-S1(9)/PW(21)-S2(7)/PW(18)-S3(5)/PW(25) Q2-S5(5)/PW(25) Q4-S11(9)/PW(19)-S12(4)/PW(24)-S13(5)/PW(15) Q5-S14(3)/PW(23)-S15(4)/PW(24) Q6-S20(8)/PW(18)
	5	Q1-S1(9)/PW(20)-S2(9)/PW(23)-S3(4)/PW(24) Q3-S8(6)/PW(14) Q4-S12(9)/PW(15)-S13(6)/PW(16) Q5-S14(4)/PW(11)-S15(4)/PW(24) Q6-S18(3)/PW(17)-S20(8)/PW(16)
37	1	Q1-S3(8)/PW(15) Q2-S6(5)/PW(24) Q3-S7(4)/PW(14) Q4-S9(4)PW(20)-S10(5)/PW(18) Q5-S13(8)/PW(18)-S14(9)/PW(19) Q6-S19(4)/PW(24)
	2	Q1-S1(2)/PW(20) Q2-S6(8)/PW(28) Q4-S11(8)/PW(18)-S13(3)/PW(23) Q5-S14(7)/PW(27) Q6-S18(8)/PW(18)-S19(2)/PW(22)
	3	Q1-S2(9)/PW(25)-S3(3)/PW(23) Q2-S6(6)/PW(16) Q3-S7(4)/PW(24)-S8(7)/PW(17) Q4-S8(3)/PW(8) Q5-S16(4)/PW(14) Q6-S20(7)/PW(17)
	4	Q1-S1(5)/PW(18)-S2(7)/PW(17)-S3(3)/PW(23) Q2-S6(8)/PW(8) Q4-S11(8)/PW(8)-S13(3)/PW(23) Q5-S14(7)/PW(17) Q6-S18(8)/PW(18)-S19(2)/PW(22)
	5	Q1-S2(5)/PW(25)-S3(3)/PW(23) Q3-S7(4)/PW(24)-S8(7)/PW(17) Q4-S8(8)/PW(20) Q5-S15(5)/PW(22)-S16(4)/PW(24) Q6-S20(7)/PW(27)
38	1	Q1-S1(2)/PW(19)-S3(14)/PW(17) Q2-S6(7)/PW(19) Q4-S10(7)/PW(20)-S12(2)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(25)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(24) Q6-S19(3)/PW(13)-S20(4)/PW(24)
	3	Q1-S1(3)/PW(19)-S2(2)/PW(11)-S3(13)/PW(13) Q2-S5(5)/PW(15) Q3-S7(6)/PW(17) Q4-S15(7)/PW(19)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(1)/PW(19)-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(19) Q6-S20(7)/PW(24)
	5	Q1-S1(3)/PW(10)-S2(7)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(19)-S9(6)/PW(16) Q6-S20(10)/PW(21)
39	1	Q1-S1(1)/PW(24)-S2(8)/PW(26)-S3(4)/PW(14) Q2-S4(5)/PW(18)-S6(7)/PW(17) Q4-S10(7)/PW(17)-S12(2)/PW(18) Q5-S15(8)/PW(18) Q6-S19(4)/PW(19)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(22)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(3)/PW(17)-S2(2)/PW(22)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(3)/PW(21)-S5(5)7PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(26) Q6-S20(7)/PW(17)
	5	Q1-S1(4)/PW(21)-S2(9)/PW(27) Q2-S5(8)/PW(18) Q3-S8(4)/PW(24)-S9(6)/PW(26) Q6-S20(5)/PW(27)
40	1	Q1-S1(2)/PW(14)-S2(5)/PW(15)-S3(4)/PW(19) Q2-S6(8)/PW(18)-S7(9)/PW(19) Q4-S10(5)/PW(15)-S12(2)/PW(12) Q5-S15(6)/PW(16)-S16(5)/PW(15) Q6-S20(5)/PW(15)
	2	Q1-S1(13)/PW(23)-S2(7)/PW(27) Q2-S4(9)/PW(19)-S5(8)/PW(18)-S6(6)/PW(16) Q3-S8(6)/PW(26)-S9(7)/PW(7) Q5-S16(4)/PW(24)-S17(5)/PW(25) Q6-S18(5)/PW(15)-S19(3)/PW(23) S20(4)/PW(14)
	3	Q1-S1(8)/PW(18)-S3(3)/PW(28) Q2-S5(9)/PW(19) Q3-S7(7)/PW(17) Q4-S15(9)/PW(19)-S16(8)/PW(18)
	4	Q1-S1(3)/PW(19)-S2(8)/PW(18)-S3(5)/PW(25) Q2-S4(4)/PW(21)-S5(7)/PW(17)-S6(6)/PW(26) Q3-S8(4)/PW(16)-S9(5)/PW(25) Q6-S19(4)/PW(24)-S20(7)/PW(17)
	5	Q1-S1(5)/PW(15)-S2(6)/PW(16) Q2-S5(8)/PW(28) Q3-S8(8)/PW(18)-S9(6)/PW(16) Q6-S19(9)/PW(17)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
41	1	Q1-S1(8)/PW(16) Q2-S4(5)/PW(18) Q3-S7(3)/PW(13) Q4-S11(4)/PW(14)-S12(7)/PW(17) Q5-S14(8)/PW(18)-S15(4)/PW(14) Q6-S20(8)/PW(18)
	2	Q1-S2(3)/PW(23)-S3(5)/PW(15) Q2-S5(7)/PW(17) Q4-S11(9)/PW(19)-S12(7)/PW(19)-S13(5)/PW(15) Q5-S14(8)/PW(18)-S15(4)/PW(14) Q6-S18(4)/PW(24) Q6-S20(8)/PW(18)
	3	Q1-S1(6)/PW(16)-S2(5)/PW(19)-S3(4)/PW(14) Q3-S8(2)/PW(22) Q4-S12(9)/PW(17)-S13(6)/PW(26) Q5-S14(8)/PW(18)-S15(4)/PW(21) Q6-S18(9)/PW(19)-S20(8)/PW(18)
	4	Q1-S1(9)/PW(19)-S2(8)/PW(18)-S3(5)PW(25) Q2-S5(5)/PW(25) Q4-S11(9)/PW(18)-S12(4)/PW(24)-S13(5)/PW(15) Q5-S14(13)/PW(23)-S15(4)/PW(24) Q6-S20(8)/PW(18)
	5	Q1-S1(6)/PW(19)-S2(9)/PW(19)-S3(4)/PW(24) Q3-S8(6)/PW(26) Q4-S12(11)/PW(19)-S13(6)/PW(16) Q5-S14(8)/PW(18)-S15(4)/PW(24) Q6-S18(3)/PW(14)-S20(8)/PW(18)
42	1	Q1-S3(8)/PW(19) Q2-S6(5)/PW(25) Q3-S7(4)/PW(14) Q4-S9(4)PW(14)-S10(5)/PW(15) Q5-S13(8)/PW(18)-S14(7)/PW(19) Q6-S19(4)/PW(24)
	2	Q1-S1(8)/PW(20) Q2-S6(8)/PW(18) Q4-S11(8)/PW(18)-S13(3)/PW(23) Q5-S14(7)/PW(27) Q6-S18(8)/PW(18)-S19(2)/PW(22
	3	Q1-S2(7)/PW(25)-S3(3)/PW(23) Q2-S6(3)/PW(16) Q3-S7(4)/PW(24)-S8(7)/PW(17) Q4-S8(8)/PW(8) Q5-S16(4)/PW(14) Q6-S20(7)/PW(17)
	4	Q1-S1(6)/PW(18)-S2(7)/PW(17)-S3(3)/PW(23) Q2-S6(8)/PW(8) Q4-S11(8)/PW(8)-S13(3)/PW(23) Q5-S14(7)/PW(17) Q6-S18(8)/PW(18)-S19(2)/PW(22)
	5	Q1-S2(5)/PW(15)-S3(3)/PW(23) Q3-S7(4)/PW(24)-S8(7)/PW(17) Q4-S8(8)/PW(18) Q5-S15(5)/PW(25)-S16(4)/PW(24) Q6-S20(5)/PW(17)
43	1	Q1-S1(8)/PW(19)-S3(4)/PW(14) Q2-S6(7)/PW(17) Q4-S10(7)/PW(17)-S12(1)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(13) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(26) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(2)/PW(17)-S2(2)/PW(21)-S3(3)/PW(15) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(5)/PW(19)-S5(3)/PW(18) Q3-S8(3)/PW(19)-S9(6)/PW(16) Q6-S20(7)/PW(27)
	5	Q1-S1(4)/PW(21)-S2(2)/PW(17) Q2-S5(8)/PW(18) Q3-S8(3)/PW(15)-S9(9)/PW(16) Q6-S20(2)/PW(27)
44	1	Q1-S1(1)/PW(20)-S2(5)/PW(22)-S3(4)/PW(14) Q2-S4(5)/PW(15)-S6(7)/PW(27) Q4-S10(7)/PW(17)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(24)
	3	Q1-S1(4)/PW(17)-S2(2)/PW(12)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(25)-S2(5)/PW(17) Q2-S4(1)/PW(21)-S5(3)PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(26) Q6-S20(7)/PW(17)
	5	Q1-S1(3)/PW(16)-S2(5)/PW(17) Q2-S5(8)/PW(18) Q3-S8(3)/PW(24)-S9(6)/PW(26) Q6-S20(3)/PW(17)
45	1	Q1-S1(9)/PW(14)-S2(5)/PW(15)-S3(4)/PW(13) Q2-S6(8)/PW(18)-S7(6)/PW(12) Q4-S10(5)/PW(15)-S12(2)/PW(12) Q5-S15(6)/PW(16)-S16(5)/PW(15) Q6-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(6)/PW(19)-S5(8)/PW(18)-S6(6)/PW(16) Q3-S8(6)/PW(16)-S9(9)/PW(7) Q5-S16(4)/PW(24)-S17(5)/PW(25) Q6-S18(5)/PW(15)-S19(3)/PW(23)-S20(4)/PW(14)
	3	Q1-S1(8)/PW(18)-S3(3)/PW(23) Q2-S5(6)/PW(19) Q3-S7(7)/PW(17) Q4-S15(9)/PW(19)-S16(8)/PW(18)
	4	Q1-S1(6)/PW(16)-S2(8)/PW(18)-S3(5)/PW(25) Q2-S4(4)/PW(24)-S5(3)/PW(17)-S6(6)/PW(26) Q3-S8(6)/PW(16)-S9(5)/PW(25) Q6-S19(4)/PW(24)-S20(5)/PW(17)
	5	Q1-S1(5)/PW(15)-S2(6)/PW(16) Q2-S5(8)/PW(18) Q3-S8(8)/PW(21)-S9(6)/PW(16) Q6-S19(9)/PW(20)
46	1	Q1-S1(4)/PW(17)-S2(2)/PW(15)-S3(8)/PW(18) Q2-S6(7)/PW(17)-S7(5)/PW(26) Q4-S10(7)/PW(17)-S12(2)/PW(22) Q5-S15(8)/PW(28) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(5)/PW(19)-S2(2)/PW(22)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(14)
	4	Q1-S1(4)/PW(19)-S2(7)/PW(17) Q2-S4(2)/PW(19)-S5(8)/PW(18) Q3-S8(3)/PW(16)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S1(3)/PW(18)-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S8(12)/PW(4)-S9(6)/PW(6) Q6-S20(3)/PW(16)
47	1	Q1-S3(4)/PW(24) Q2-S5(8)/PW(8)-S6(7)/PW(17) Q4-S10(7)/PW(17)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S19(14)/PW(4)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(5)-S5(8)/PW(8) Q3-S8(6)/PW(6) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(7)/PW(17)-S2(9)/PW(18)-S3(3)/PW(13) Q2-S5(5)/PW(5) Q3-S7(6)/PW(6) Q4-S15(7)/PW(17)
	4	Q1-S1(6)/PW(13)-S2(7)/PW(17) Q2-S4(1)/PW(11)-S5(5)/PW(8) Q3-S8(3)/PW(16)-S9(6)/PW(21) Q6-S20(7)/PW(22)
	5	Q1-S1(3)/PW(12)-S2(7)/PW(17) Q2-S5(9)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q6-S20(4)/PW(11)
48	1	Q1-S1(6)/PW(19)-S3(8)/PW(18) Q2-S6(4)/PW(14) Q4-S10(7)/PW(17)-S12(2)/PW(12) Q5-S15(8)/PW(8) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(6)/PW(21) Q2-S4(6)/PW(16)-S5(8)/PW(18) Q3-S8(6)/PW(19) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(7)/PW(17)-S2(5)/PW(15)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)PW(13)-S2(7)/PW(17) Q2-S4(10)/PW(11)-S5(8)/PW(19) Q3-S8(4)/PW(16)-S9(6)/PW(16) Q6-S20(11)/PW(16)
	5	Q1-S1(4)/PW(14)-S2(7)/PW(17) Q2-S5(8)/PW(22) Q3-S8(6)/PW(14)-S9(3)/PW(16) Q6-S20(8)/PW(21)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
49	1	Q1-S3(6)/PW(11) Q2-S5(8)/PW(18)-S6(7)/PW(17)-S7(9)/PW(19) Q4-S11(6)/PW(9)-S12(4)/PW(12) Q5-S15(8)/PW(18) Q6-S19(5)/PW(13)-S20(5)/PW(11)
	2	Q1-S1(6)/PW(8)-S2(9)/PW(9) Q2-S4(5)/PW(15)-S5(8)/PW(8) Q3-S7(7)/PW(7)-S8(6)/PW(6) Q5-S16(4)/PW(14)-S17(6)/PW(6) Q6-S20(4)/PW(14)
	3	Q1-S1(6)/PW(17)-S3(3)/PW(13) Q2-S5(5)/PW(15)-S6(8)/PW(18) Q3-S7(6)/PW(22)-S8(8)/PW(8) Q4-S15(7)/PW(17)-S16(9)/PW(9)
	4	Q1-S1(3)/PW(21)-S2(7)/PW(17)-S3(4)/PW(18) Q2-S4(3)/PW(11)-S5(6)/PW(18) Q3-S8(6)/PW(14)-S9(6)/PW(13) Q6-S20(7)/PW(17)
	5	Q1-S1(1)/PW(15)-S2(7)/PW(11)-S3(7)/PW(17) Q2-S5(8)/PW(8) Q3-S8(4)/PW(22)-S9(6)/PW(16) Q6-S19(8)/PW(18)-S20(7)/PW(21)
50	1	Q2-S5(4)/PW(14)-S6(5)/PW(15) Q4-S10(7)/PW(17)-S11(8)/PW(18)-S12(2)/PW(22) Q5-S15(8)/PW(21)-S17(9)/PW(19) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(7)/PW(23)-S2(5)/PW(25) Q2-S4(5)/PW(25)-S5(7)/PW(28)-S6(4)/PW(24) Q3-S8(4)/PW(14) Q5-S16(2)/PW(14)-S17(5)/PW(15) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(6)/PW(16)-S2(8)/PW(18)-S3(3)/PW(13) Q2-S5(5)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(16)-S8(8)/PW(18) Q4-S15(7)/PW(17)
	4	Q1-S1(4)/PW(15)-S2(7)/PW(17) Q2-S4(4)/PW(14)-S5(8)/PW(21) Q3-S8(5)/PW(16)-S9(6)/PW(16) Q6-S18(5)/PW(18)-S20(7)/PW(17)
	5	Q1-S1(4)/PW(14)-S2(6)/PW(17)-S3(8)/PW(18) Q2-S5(8)/PW(18) Q3-S8(5)/PW(14)-S9(6)/PW(26) Q6-S19(7)/PW(17)
51	1	Q1-S2(9)/PW(18) Q2-S5(4)/PW(14)-S6(7)7PW(17) Q4-S10(4)/PW(14)-S11(8)/PW(18)-S12(5)/PW(15) Q5-S15(8)/PW(8)-S16(7)/PW(7) Q6-S18(6)/PW(16)-S19(8)/PW(18)-S20(9)/PW(21)
	2	Q1-S1(8)/PW(10)-S2(8)/PW(12) Q2-S4(5)/PW(17)-S5(8)/PW(20)-S6(9)/PW(19) Q3-S8(6)/PW(16) Q5-S16(3)/PW(14) Q6-S19(3)/PW(11)-S20(4)/PW(14)
	3	Q1-S2(5)/PW(15) Q2-S6(6)/PW(15)-S7(7)/PW(17) Q4-S15(7)/PW(16)
	4	Q1-S1(8)/PW(17)-S2(4)/PW(19)-S3(9)/PW(9) Q2-S4(8)/PW(21)-S5(5)/PW(10) Q3-S8(5)/PW(15)-S9(6)/PW(16) Q5-S18(4)/PW(14) Q6-S19(8)/PW(18)-S20(7)/PW(17)
	5	Q1-S1(5)/PW(16)-S2(7)/PW(17) Q2-S4(7)/PW(17)-S5(8)/PW(12) Q3-S7(5)/PW(15)-S8(4)/PW(24)-S9(6)/PW(16) Q6-S19(8)/PW(20)-S20(6)/PW(16)
52	1	Q1-S3(8)/PW(23) Q2-S6(6)/PW(16)-S7(5)/PW(15) Q4-S10(3)/PW(14)-S11(3)/PW(13)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S20(5)/PW(15)
	2	Q1-S1(5)/PW(21)-S2(6)/PW(18) Q2-S4(6)/PW(24)-S5(7)/PW(17) Q3-S8(7)/PW(14) Q5-S16(5)/PW(15)-S17(4)/PW(14) Q6-S19(7)/PW(17)-S20(8)/PW(18)
	3	Q1-S1(4)/PW(22)-S2(6)/PW(18) Q2-S5(7)/PW(17)-S6(5)/PW(24) Q3-S7(6)/PW(11)-S8(5)/PW(15) Q4-S15(7)/PW(19)-S16(8)/PW(18)
	4	Q1-S1(4)/PW(18)-S2(6)/PW(16) Q2-S4(5)/PW(15)-S5(4)/PW(24) Q3-S8(7)/PW(24) Q6-S20(7)/PW(21)
	5	Q1-S2(8)/PW(8) Q2-S5(7)/PW(18) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q6-S19(9)/PW(19)
53	1	Q1-S2(9)/PW(18)-S3(3)/PW(12) Q2-S4(6)/PW(16)-S6(7)/PW(17) Q4-S10(6)/PW(20)-S11(8)/PW(18)-S12(2)/PW(12) Q5-S15(7)/PW(18)-S16(9)/PW(9) Q6-S18(7)/PW(17)-S19(6)/PW(16)-S20(5)/PW(20)
	2	Q1-S1(8)/PW(15)-S2(6)/PW(26) Q2-S4(5)/PW(15)-S5(8)/PW(18)-S6(7)/PW(17) Q3-S8(6)/PW(24)-S9(7)/PW(17) Q5-S16(4)/PW(24)-S17(5)/PW(15) Q6-S19(6)/PW(16)
	3	Q1-S3(3)/PW(13) Q2-S5(5)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S15(7)/PW(17)-S16(6)/PW(16)
	4	Q1-S2(7)/PW(17) Q2-S5(8)/PW(18) Q3-S8(5)/PW(15)-S9(6)/PW(16)-S10(4)/PW(24) Q6-S19(8)/PW(18)
	5	Q1-S1(1)/PW(24)-S2(7)/PW(17) Q2-S5(3)/PW(21) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q6-S18(4)/PW(22)-S19(8)/PW(19)
54	1	Q2-S6(9)/PW(17) Q4-S10(7)/PW(21)-S12(1)/PW(22) Q5-S15(10)/PW(18) Q6-S19(11)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(4)/PW(24) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(4)/PW(17)-S2(2)/PW(22)-S3(3)/PW(13) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(12)-S2(6)(PW(14) Q2-S4(2)/PW(11)-S5(8)/PW(18) Q3-S8(6)/PW(23)-S9(6)/PW(16) Q6-S20(7)/PW(17)
	5	Q1-S1(1)/PW(21)-S2(5)/PW(17) Q2-S5(3)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q6-S20(7)/PW(22)
55	1	Q1-S1(5)/PW(18) Q2-S6(7)/PW(17)-S7(5)/PW(15) Q3-S15(8)/PW(8) Q4-S10(4)/PW(17)-S11(5)/PW(15)-S12(2)/PW(23) Q5-S15(8)/PW(8) Q6-S18(9)/PW(19)-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q3-S8(6)/PW(16) Q5-S16(4)/PW(14) Q6-S19(3)/PW(13)-S20(4)/PW(14)
	3	Q1-S1(11)/PW(19)-S3(3)/PW(23) Q2-S5(5)/PW(25) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(6)/PW(26)-S2(7)/PW(11)-S3(8)/PW(18) Q2-S4(9)/PW(15)-S5(8)/PW(18) Q3-S8(8)/PW(16)-S9(6)/PW(19)-S10(7)/PW(17) Q6-S19(6)/PW(16)-S20(9)/PW(16)
	5	Q1-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S9(6)/PW(16) Q6-S19(9)/PW(24)-S20(7)/PW(17)
56	1	Q4-S10(7)/PW(15)-S12(9)/PW(20) Q5-S15(8)/PW(18)
	2	Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(5)/PW(24) Q6-S19(3)/PW(13)-S20(4)/PW(24)
	3	Q1-S1(9)/PW(18)-S2(2)/PW(24)-S3(3)/PW(23) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(18)-S2(7)/PW(17) Q2-S4(3)/PW(19)-S5(8)/PW(21) Q3-S8(10)/PW(21)-S9(6)/PW(15) Q6-S20(7)/PW(18)
	5	Q1-S1(4)/PW(13)-S2(9)/PW(20)-S3(9)/PW(15) Q2-S5(8)/PW(19) Q3-S6(4)/PW(16)-S8(7)/PW(18)-S9(6)/PW(16) Q6-S19(5)/PW(17)-S20(4)/PW(21)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
57	1	Q1-S2(9)/PW(21)-S3(5)/PW(15) Q2-S6(4)/PW(17) Q4-S10(7)/PW(17)-S12(5)/PW(22) Q5-S15(8)/PW(18)-S16(5)/PW(25) Q6-S18(6)/PW(26)-S20(5)/PW(25)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S4(5)/PW(15) Q3-S7(4)/PW(14)-S8(6)/PW(16) Q5-S16(4)/PW(24)-S17(3)/PW(17) Q6-S20(9)/PW(19)
	3	Q1-S1(8)/PW(19)-S3(4)/PW(14) Q2-S5(5)/PW(15)-S6(7)/PW(17) Q3-S7(6)/PW(26)-S8(5)/PW(15) Q4-S15(7)/PW(17)-S16(8)/PW(18) Q5-S18(9)/PW(19)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(1)/PW(21)-S5(4)/PW(8) Q3-S8(6)/PW(13)-S9(6)/PW(19) Q6-S20(7)/PW(22)
	5	Q1-S2(9)/PW(21) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16)
58	1	Q1-S2(9)/PW(18)-S3(4)/PW(14) Q2-S6(7)/PW(17) Q4-S10(7)/PW(27)-S12(9)/PW(19) Q5-S15(8)/PW(18) Q6-S18(5)PW(15)-S19(4)/PW(24)-S20(5)/PW(25)
	2	Q1-S1(8)/PW(19)-S2(5)/PW(25) Q2-S4(9)/PW(24) Q3-S8(6)/PW(26)-S9(6)/PW(16) Q5-S16(6)/PW(16)-S17(7)/PW(17) Q6-S19(8)/PW(18)-S20(4)/PW(24)
	3	Q1-S1(9)/PW(19)-S3(7)/PW(17) Q2-S5(5)PW(15) Q3-S7(6)/PW(16)-S8(5)/PW(15) Q4-S15(9)/PW(19)-S16(4)/PW(24)
	4	Q1-S2(9)/PW(9)-S3(4)/PW(14) Q2-S5(3)/PW(13)-S6(5)/PW(15) Q3-S8(6)/PW(16) Q6-S18(3)/PW(13)-S19(8)/PW(18)-S20(4)/PW(14)
	5	Q1-S1(6)/PW(19)-S2(7)/PW(16)-S3(5)/PW(15) Q2-S5(8)/PW(8)-S8(4)/PW(14) Q3-S8(4)/PW(14)-S9(6)/PW(26)-S10(5)/PW(15) Q6-S19(9)/PW(18)-S20(7)/PW(13)
59	1	Q2-S6(3)/PW(14)-S7(8)/PW(18) Q4-S10(7)/PW(17)-S11(5)/PW(25)-S12(2)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(6)/PW(16) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(26) Q5-S16(5)/PW(26) Q6-S19(13)/PW(13)-S20(4)/PW(14)
	3	Q1-S2(9)/PW(9)-S3(7)/PW(18) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S2(9)/PW(19) Q2-S4(11)/PW(11)-S5(8)/PW(18) Q3-S8(4)/PW(7)-S9(6)/PW(16) Q6-S19(2)/PW(22)-S20(3)/PW(14)
	5	Q2-S5(8)/PW(8) Q3-S8(4)/PW(24)-S9(6)/PW(16) Q6-S19(5)/PW(15)-S20(6)/PW(19)
60	1	Q2-S6(7)/PW(11)-S7(9)/PW(9) Q3-S9(8)PW(15) Q4-S10(3)/PW(19)-S11(8)/PW(18)-S12(2)/PW(22) Q5-S15(9)/PW(9) Q6-S18(6)/PW(16)-S19(4)/PW(14)-S20(5)/PW(25)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S4(9)/PW(9)-S5(6)/PW(16) Q3-S7(7)/PW(17)-S8(6)/PW(16) Q6-S20(4)/PW(14)
	3	Q1-S2(12)/PW(12) Q2-S5(8)/PW(9) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S15(8)/PW(19)-S16(9)/PW(9) Q6-S19(8)/PW(18)
	4	Q1-S2(7)/PW(17)-S3(8)/PW(18) Q3-S8(6)/PW(16)-S9(6)/PW(16)-S10(5)/PW(25) Q5-S17(9)/PW(19) Q6-S19(5)/PW(25)-S20(8)/PW(18)
	5	Q2-S5(3)/PW(18) Q3-S8(5)/PW(21)-S9(6)/PW(16)
61	1	Q1-S1(6)/PW(21)-S2(5)/PW(19)-S3(4)/PW(14) Q4-S10(5)/PW(15)-S11(7)/PW(16)-S12(8)/PW(22) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q2-S4(5)/PW(25)-S5(8)/PW(18) Q3-S8(6)/PW(26)-S9(7)/PW(17) Q6-S19(9)/PW(9)
	3	Q1-S3(5)/PW(9) Q2-S5(5)/PW(25)-S6(7)/PW(17) Q3-S7(6)/PW(26) Q4-S15(7)/PW(17) Q6-S18(8)/PW(18)-S19(9)/PW(19)
	4	Q1-S2(7)/PW(17) Q2-S4(11)/PW(11)-S5(8)/PW(18) Q3-S8(6)/PW(19)-S9(6)/PW(18) Q6-S20(7)/PW(20)
	5	Q1-S1(4)/PW(24)-S2(5)/PW(16) Q2-S5(8)/PW(27) Q3-S8(4)/PW(22)-S9(6)/PW(16) Q4-S11(4)/PW(17)
62	1	Q1-S1(8)/PW(18)-S3(7)/PW(24) Q2-S5(6)/PW(18) Q4-S10(7)/PW(17)-S11(8)/PW(9)-S12(5)/PW(17) Q5-S15(9)/PW(8)-S16(3)/PW(15) Q6-S20(7)/PW(9)
	2	Q1-S1(6)/PW(3)-S2(8)/PW(9) Q3-S8(4)/PW(16)-S9(5)/PW(8) Q5-S16(8)/PW(14)-S17(5)/PW(15) Q6-S19(7)/PW(9)
	3	Q1-S1(6)/PW(18)-S2(7)/PW(12) Q2-S5(7)/PW(9) Q3-S7(8)/PW(16)-S8(4)/PW(19) Q4-S15(7)/PW(27)-S16(6)/PW(18)
	4	Q2-S5(9)/PW(8)-S6(3)/PW(7) Q3-S8(9)/PW(6)-S9(5)/PW(9) Q6-S18(3)/PW(3)-S20(9)/PW(9)
	5	Q1-S2(5)/PW(17)-S3(9)/PW(9) Q2-S4(2)/PW(20)-S5(8)/PW(18) Q3-S8(5)/PW(21)-S9(6)/PW(16)-S10(3)/PW(13) Q4-S16(4)/PW(14) Q5-S17(3)/PW(18)-S18(5)/PW(15) Q6-S20(9)/PW(21)
63	1	Q1-S3(9)/PW(24) Q2-S6(7)/PW(17) Q5-S15(8)/PW(18)-S16(9)/PW(9)
	2	Q1-S1(7)/PW(23)-S2(3)/PW(18) Q2-S4(5)/PW(15)-S5(8)/PW(18)-S6(7)/PW(17) Q3-S8(6)/PW(16) Q4-S15(6)/PW(16) Q6-S20(9)/PW(19)
	3	Q1-S1(7)/PW(9)-S2(8)/PW(8) Q2-S4(4)/PW(24)-S5(5)/PW(25) Q3-S7(6)/PW(26)-S8(9)/PW(17)
	4	Q2-S4(3)/PW(23)-S5(7)/PW(17)-S6(3)/PW(18) Q6-S19(9)/PW(23)-S20(7)/PW(11)
	5	Q2-S5(7)/PW(19) Q3-S9(6)/PW(6) Q6-S18(8)/PW(9)-S20(5)/PW(15)
64	1	Q1-S1(9)/PW(14)-S2(3)/PW(20)-S3(2)/PW(14) Q2-S5(9)/PW(16)-S6(7)/PW(17) Q4-S10(2)/PW(19)-S11(8)/PW(18) Q5-S14(6)/PW(16)-S15(8)/PW(18) Q6-S19(6)/PW(16)-S20(3)/PW(13)
	2	Q2-S5(6)/PW(16)-S6(7)/PW(17) Q3-S8(9)/PW(19)-S9(5)/PW(15) Q5-S16(7)/PW(23)-S17(2)/PW(22) Q6-S20(4)/PW(12)
	3	Q1-S1(6)/PW(16)-S3(9)/PW(9) Q4-S15(7)/PW(21)-S16(5)/PW(15) Q6-S19(8)/PW(18)
	4	Q1-S1(3)/PW(13)-S2(7)/PW(17)-S3(6)/PW(16) Q2-S5(8)/PW(8)-S6(5)/PW(15) Q6-S19(8)/PW(18)-S20(9)/PW(9)
	5	Q1-S2(3)/PW(17) Q2-S5(9)/PW(14)-S6(5)/PW(15) Q3-S8(7)/PW(21)
65	1	Q1-S1(7)/PW(15)-S2(4)/PW(16)-S3(5)/PW(14) Q2-S5(8)PW(18)-S6(7)/PW(17)-S7(5)/PW(15) Q3-S8(3)/PW(23) Q4-S12(9)/PW(9) Q5-S15(8)/PW(18)
	2	Q1-S1(9)/PW(9)-S2(5)/PW(15) Q2-S4(4)/PW(24)-S5(7)/PW(17) Q3-S8(5)/PW(15)-S9(4)PW(24) Q5-S16(8)/PW(8) Q6-S19(9)/PW(9)-S20(6)/PW(6)
	3	Q1-S2(7)/PW(18)-S3(9)/PW(9) Q2-S5(5)/PW(15) Q3-S6(4)/PW(10)-S7(6)/PW(16)
	4	Q1-S1(5)/PW(16)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S4(9)/PW(12)-S5(8)/PW(18)-S6(3)/PW(11) Q3-S8(6)/PW(16)-S9(6)/PW(16) Q6-S19(8)/PW(18)-S20(7)/PW(17)
	5	Q1-S2(8)/PW(17) Q2-S4(5)/PW(14)-S5(3)/PW(8) Q3-S8(7)/PW(14)-S9(6)/PW(16) Q6-S19(6)/PW(16)-S20(7)/PW(17)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
66	1	Q1-S3(4)/PW(14) Q2-S6(7)/PW(17)-S7(8)/PW(18) Q4-S10(7)/PW(17) Q5-S15(8)/PW(8)-S16(8)/PW(8) Q6-S18(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(7)/PW(17)-S2(8)/PW(18) Q2-S4(2)/PW(21)-S5(3)/PW(13)-S6(4)/PW(24) Q3-S7(5)/PW(15)-S8(6)/PW(16)-S9(7)/PW(17) Q5-S16(4)/PW(14)-S17(5)/PW(15) Q6-S18(7)/PW(17)-S20(4)/PW(24)
	3	Q1-S1(8)/PW(18)-S2(9)/PW(19) Q2-S5(5)/PW(15)-S6(6)/PW(16) Q3-S7(6)/PW(16)-S8(7)/PW(17) Q4-S14(7)/PW(17)-S15(8)/PW(18)
	4	Q2-S4(7)/PW(18)-S5(8)/PW(18) Q3-S8(9)/PW(19)
	5	Q1-S2(5)/PW(14) Q2-S5(8)/PW(16)-S6(7)/PW(17) Q3-S8(5)/PW(15)-S9(7)/PW(17) Q5-S16(6)/PW(16)-S17(3)/PW(23) Q6-S18(4)/PW(14)-S19(8)/PW(18)
67	1	Q1-S1(7)/PW(19)-S2(1)/PW(21)-S3(4)/PW(14) Q2-S4(3)/PW(23)-S6(5)/PW(15) Q4-S10(7)/PW(15)-S11(6)/PW(16)-S12(2)/PW(12) Q6-S19(7)/PW(17)
	2	Q1-S1(11)/PW(13)-S2(5)/PW(15)-S3(5)/PW(16) Q2-S5(9)/PW(15)-S6(7)/PW(8) Q3-S7(4)/PW(17)-S8(5)/PW(16) Q5-S6(6)/PW(4)-S17(8)PW(5)
	3	Q1-S1(6)/PW(16)-S3(8)/PW(23) Q2-S5(8)/PW(18)-S6(9)/PW(9) Q3-S7(7)/PW(17)-S8(8)/PW(19) Q4-S14(6)/PW(26)-S15(7)/PW(7) Q6-S19(8)/PW(18)-S20(6)/PW(14)
	4	Q1-S1(3)/PW(3)-S2(6)/PW(6)-S3(4)/PW(4) Q4-S5(8)/PW(8) Q6-S20(7)/PW(7)
	5	Q1-S1(6)/PW(24)-S2(7)/PW(17)-S3(8)/PW(18) Q2-S4(5)/PW(25)-S5(8)/PW(18) Q3-S8(3)/PW(24)-S9(6)/PW(16) Q5-S17(6)/PW(16) Q6-S19(4)/PW(21)-S20(7)/PW(17)
68	1	Q1-S1(7)/PW(22)-S3(6)/PW(14) Q2-S6(9)/PW(9) Q4-S10(6)/PW(16)-S12(3)/PW(13) Q5-S15(8)/PW(18) Q6-S19(6)/PW(16)-S20(8)/PW(18)
	2	Q1-S1(6)/PW(16) Q2-S4(6)/PW(22)-S5(8)/PW(18) Q3-S8(9)/PW(17) Q5-S16(6)/PW(6) Q6-S19(7)/PW(15)-S20(3)/PW(23)
	3	Q1-S1(9)/PW(19)-S3(6)/PW(26) Q2-S5(5)/PW(5) Q3-S7(7)/PW(7) Q4-S15(8)/PW(8)
	4	Q1-S1(4)/PW(14)-S2(8)/PW(18) Q2-S4(3)/PW(16)-S5(11)/PW(11) Q3-S8(9)PW(19)-S9(6)/PW(26) Q6-S19(4)/PW(14)-S20(8)/PW(18)
	5	Q1-S1(6)/PW(16)-S2(8)/PW(18) Q2-S5(8)(PW(18) Q3-S8(9)/PW(11)-S9(5)/PW(15) Q6-S19(3)/PW(23)-S20(6)/PW(16)
69	1	Q1-S1(7)/PW(17)-S3(5)/PW(13) Q2-S6(9)/PW(8) Q4-S12(9)/PW(9) Q6-S20(6)/PW(16)
	2	Q2-S4(9)/PW(11)-S5(6)/PW(16)-S6(5)/PW(15) Q3-S7(7)/PW(17)-S8(8)/PW(8) Q4-S9(9)/PW(9) Q5-S16(3)/PW(23)-S17(2)/PW(20) Q6-S20(4)/PW(9)
	3	Q1-S2(6)/PW(9)-S3(3)/PW(13) Q3-S7(7)/PW(17)-S8(8)/PW(18) Q6-S18(9)/PW(9)
	4	Q1-S2(9)/PW(17) Q2-S5(7)/PW(18)-S6(4)/PW(14) Q3-S7(6)/PW(15)-S8(6)/PW(16)-S9(6)/PW(16) Q6-S19(5)/PW(15)-S20(6)/PW(16)
	5	Q1-S2(7)/PW(17) Q2-S5(8)/PW(18) Q3-S8(4)/PW(14)-S9(6)/PW(16) Q5-S16(4)/PW(14)-S17(5)PW(15) Q6-S19(3)/PW(13)-S20(8)/PW(19)
70	1	Q1-S1(10)/PW(17)-S2(2)/PW(11)-S3(4)/PW(15) Q2-S5(6)/PW(16)-S7(7)/PW(17) Q3-S8(5)/PW(15)-S9(4)/PW(24) Q5-S15(6)/PW(16) Q6-S20(9)/PW(19)
	2	Q1-S1(3)/PW(14)-S2(5)/PW(15)-S3(8)/PW(18) Q2-S4(4)/PW(19)-S5(3)/PW(13) Q3-S7(3)/PW(23)-S8(9)/PW(9) Q5-S16(6)/PW(16)-S17(7)/PW(17)
	3	Q1-S2(13)/PW(17)-S3(3)PW(13) Q2-S5(6)/PW(16)-S6(7)/PW(17) Q3-S7(11)/PW(11) Q4-S15(3)/PW(23)-S16(4)/PW(14) Q6-S18(5)/PW(15)-S19(6)/PW(16)
	4	Q1-S1(9)/PW(9)-S2(7)/PW(17) Q2-S5(8)/PW(8) Q3-S8(5)/PW(15)-S9(4)/PW(14) Q6-S19(8)/PW(8)-S20(9)/PW(9)
	5	Q1-S2(8)/PW(9)-S3(7)/PW(17) Q2-S4(6)/PW(16)-S5(5)/PW(15) Q3-S8(4)/PW(24)-S9(3)/PW(16) Q4-S11(7)/PW(16)-S12(8)/PW(8) Q5-S16(10)/PW(19)-S17(8)/PW(18)
71	1	Q1-S1(7)/PW(18)-S3(5)PW(25) Q2-S6(6)/PW(16) Q4-S10(6)/PW(16)-S12(3)/PW(23) Q5-S15(9)/PW(19) Q6-S19/PW(17)-S20(3)/PW(13)
	2	Q1-S1(6)/PW(16) Q2-S4(9)/PW(17)-S5(8)/PW(18) Q3-S8(5)/PW(19) Q5-S16(5)/PW(25) Q6-S19(4)/PW(24)-S20(5)/PW(15)
	3	Q1-S1(6)/PW(16)-S2(3)/PW(13)-S3(5)/PW(15) Q2-S5(8)/PW(19) Q3-S7(3)/PW(17) Q4-S15(9)/PW(19)
	4	Q1-S1(7)/PW(17)-S2(3)/PW(23) Q2-S4(2)/PW(20)-S5(4)/PW(14) Q3-S8(7)/PW(22)-S9(3)/PW(23) Q6-S20(10)/PW(18)
	5	Q1-S1(3)/PW(25)-S2(3)/PW(19) Q2-S5(5)/PW(24) Q3-S8(6)/PW(23)-S9(8)/PW(17) Q6-S20(8)/PW(26)
72	1	Q3-S7(9)/PW(16) Q4-S10(8)/PW(18)-S11(6)/PW(17)-S12(3)/PW(23) Q5-S15(5)/PW(14) Q6-S20(5)/PW(15)
	2	Q1-S1(8)/PW(19)-S2(8)/PW(18)-S3(4)/PW(24) Q2-S5(3)/PW(13) Q3-S7(5)/PW(23)-S8(7)/PW(18) Q5-S16(6)/PW(16)-S17(7)/PW(17) Q6-S19(8)/PW(18)
	3	Q1-S1(7)/PW(9) Q3-S7(6)/PW(16)-S8(9)/PW(19) Q4-S14(6)/PW(26)-S15(7)/PW(17) Q5-S16(9)/PW(9)-S17(3)/PW(13) Q6-S18(8)/PW(8)-S19(9)/PW(9)
	4	Q1-S1(10)/PW(16)-S2(3)/PW(17)-S3(5)/PW(10) Q2-S4(2)/PW(5) Q3-S8(3)/PW(13)-S9(7)/PW(7) Q5-S16(5)/PW(15) Q6-S18(5)/PW(15)-S19(9)/PW(9)
	5	Q1-S2(9)/PW(9) Q3-S8(3)/PW(13)-S9(2)/PW(21) Q6-S19(5)/PW(16)-S20(3)/PW(27)
73	1	Q1-S1(6)/PW(23)-S2(2)/PW(19)-S3(4)/PW(13) Q2-S5(5)/PW(15)-S6(7)/PW(17) Q3-S7(3)/PW(16)-S8(5)/PW(20) Q4-S10(4)/PW(17)-S11(3)/PW(13)-S12(2)/PW(21) Q5-S15(8)/PW(18)-S16(4)/PW(24) Q6-S18(6)/PW(16)-S19(4)/PW(24)-S20(5)/PW(15)
	2	Q1-S1(5)/PW(13)-S2(8)/PW(18) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(7)/PW(17) Q5-S16(4)/PW(14)-S17(5)/PW(23) Q6-S18(5)/PW(15)-S20(4)/PW(24)
	3	Q1-S1(7)/PW(16)-S3(13)/PW(23) Q2-S5(6)/PW(16) Q3-S7(9)/PW(19) Q4-S15(8)/PW(18)-S16(9)/PW(19)
	4	Q1-S2(9)/PW(19)-S3(5)/PW(15) Q2-S4(3)/PW(11)-S5(9)/PW(9) Q3-S7(4)/PW(14) Q4-S10(5)/PW(15)-S11(6)/PW(16)
	5	Q1-S2(8)/PW(19)-S3(8)/PW(19) Q2-S4(4)/PW(22)-S5(8)/PW(12) Q3-S8(3)PW(13)-S9(9)/PW(16)-S10(2)/PW(14) Q5-S16(8)/PW(20)-S17(3)/PW(13) Q6-S18(7)/PW(15)-S19(6)/PW(11)-S20(3)/PW(27)

Table A1. *Cont.*

Part	Route	Cell–Machine (Operating Time)/(Machine Power Amount)
74	1	Q4-S10(8)/PW(16)-S11(8)/PW(18)-S12(2)PW(22) Q5-S15(9)/PW(18)-S16(3)/PW(18) Q6-S19(5)/PW(15)
	2	Q1-S1(3)/PW(23)-S2(8)/PW(18) Q2-S5(8)/PW(18)-S6(7)/PW(21) Q3-S8(5)/PW(22)-S9(4)/PW(18)-S10(3)/PW(13) Q5-S16(4)/PW(14)-S17(5) Q6-S18(5)/PW(17)-S20(4)/PW(20)
	3	Q1-S1(9)/PW(9)-S3(5)/PW(15) Q2-S5(5)/PW(15)-S6(6)/PW(16) Q3-S7(6)/PW(16)-S8(8)/PW(18) Q4-S15(7)/PW(17)-S16(8)/PW(18)
	4	Q1-S1(8)/PW(23)-S2(7)/PW(17)-S3(4)/PW(24) Q2-S5(8)/PW(18) Q3-S8(6)/PW(16)-S9(5)/PW(15) Q6-S19(9)/PW(8
	5	Q1-S1(6)/PW(16)-S2(7)/PW(23)-S3(6)/PW(15) Q2-S5(7)/PW(18)-S6(8)/PW(24) Q3-S8(4)/PW(14)-S9(6)/PW(25)-S10(8)/PW(23) Q6-S18(4)/PW(14)-S19(6)/PW(15)-S20(8)/PW(17)
75	1	Q2-S6(4)/PW(18)-S7(9)/PW(19) Q4-S10(6)/PW(17)-S12(5)/PW(22) Q5-S15(8)/PW(18)
	2	Q1-S1(7)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(16) Q5-S16(5)/PW(14) Q6-S19(6)/PW(23)-S20(4)/PW(14)
	3	Q1-S1(6)/PW(17)-S2(2)/PW(22)-S3(5)/PW(15) Q2-S5(4)/PW(5) Q3-S7(3)/PW(6) Q4-S15(5)/PW(7)
	4	Q1-S1(4)/PW(23)-S2(7)/PW(17) Q2-S4(5)/PW(21)-S5(6)/PW(18) Q3-S8(8)/PW(6)-S9(6)/PW(6) Q6-S20(6)/PW(17)
	5	Q1-S1(1)/PW(11)-S2(4)/PW(17) Q2-S5(3)/PW(18) Q3-S8(5)/PW(15)-S9(6)/PW(16) Q6-S20(9)/PW(25)
76	1	Q1-S1(7)/PW(24) Q2-S4(7)/PW(15) Q3-S7(4)/PW(13) Q4-S11(6)/PW(14)-S12(7)/PW(17) Q5-S14(5)/PW(18)-S15(4)/PW(14) Q6-S20(8)/PW(18)
	2	Q1-S2(3)/PW(13)-S3(5)/PW(15) Q2-S5(7)/PW(17) Q4-S11(5)/PW(19)-S12(4)/PW(17)-S13(3)/PW(15) Q5-S14(8)/PW(18)-S15(4)/PW(14) Q6-S18(4)/PW(14)-S20(8)/PW(18)
	3	Q1-S1(5)/PW(16)-S2(7)/PW(19)-S3(4)/PW(14) Q3-S8(8)/PW(22) Q4-S12(9)/PW(19)-S13(8)/PW(26) Q5-S14(8)/PW(18)-S15(4)/PW(24). Q6-S18(9)/PW(19)-S20(8)/PW(18)
	4	Q1-S1(9)/PW(19)-S2(8)/PW(18)-S3(5)/PW(25) Q2-S5(5)/PW(25) Q4-S11(9)/PW(19)-S12(4)/PW(24)-S13(5)/PW(15) Q5-S14(3)/PW(23)-S15(4)/PW(24) Q6-S20(8)/PW(28)
	5	Q1-S1(7)/PW(16)-S2(8)/PW(19)-S3(4)/PW(24) Q3-S8(6)/PW(16) Q4-S12(7)/PW(19)-S13(6)/PW(16) Q5-S14(8)/PW(18)-S15(4)/PW(24) Q6-S18(10)/PW(19)-S20(6)/PW(28)
77	1	Q1-S3(7)/PW(18) Q2-S6(5)/PW(15) Q3-S7(4)/PW(14) Q4-S9(6)PW(24)-S10(5)/PW(15) Q5-S13(7)/PW(18)-S14(9)/PW(19) Q6-S19(6)/PW(24)
	2	Q1-S1(5)/PW(20) Q2-S6(8)/PW(18) Q4-S11(8)/PW(18)-S13(3)/PW(23) Q5-S14(7)/PW(27) Q6-S18(8)/PW(18)-S19(5)/PW(22)
	3	Q1-S2(3)/PW(25)-S3(6)/PW(23) Q2-S6(4)/PW(19) Q3-S7(7)/PW(24)-S8(8)/PW(17) Q4-S8(5)/PW(8) Q5-S16(5)/PW(14) Q6-S20(7)/PW(17)
	4	Q1-S1(8)/PW(21)-S2(7)/PW(17)-S3(3)/PW(23) Q2-S6(8)/PW(8) Q4-S11(9)/PW(8)-S13(2)/PW(23) Q5-S14(7)/PW(17) Q6-S18(8)/PW(28)-S19(2)/PW(22)
	5	Q1-S2(5)/PW(15)-S3(3)/PW(23) Q3-S7(7)/PW(24)-S8(7)/PW(17) Q4-S8(8)/PW(18) Q5-S15(3)/PW(25)-S16(6)/PW(24) Q6-S20(7)/PW(24)
78	1	Q1-S1(8)/PW(22)-S3(4)/PW(14) Q2-S6(6)/PW(17) Q4-S10(8)/PW(17)-S12(2)/PW(22) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(15)-S5(8)/PW(18) Q3-S8(6)/PW(26) Q5-S16(4)/PW(18) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(6)/PW(17)-S2(2)/PW(21)-S3(3)/PW(13) Q2-S5(5)/PW(15) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(8)/PW(23)-S2(7)/PW(17) Q2-S4(6)/PW(19)-S5(6)/PW(18) Q3-S8(5)/PW(16)-S9(6)/PW(16) Q6-S20(3)/PW(17)
	5	Q1-S1(3)/PW(11)-S2(7)/PW(17) Q2-S5(8)/PW(18) Q3-S8(8)/PW(24)-S9(6)/PW(16) Q6-S20(5)/PW(26)
79	1	Q1-S1(5)/PW(11)-S2(2)/PW(22)-S3(4)/PW(14) Q2-S4(1)/PW(25)-S6(7)/PW(17) Q4-S10(7)/PW(17)-S12(2)/PW(12) Q5-S15(8)/PW(18) Q6-S19(4)/PW(14)-S20(5)/PW(15)
	2	Q1-S1(3)/PW(23) Q2-S4(5)/PW(19)-S5(8)/PW(18) Q3-S8(6)/PW(26) Q5-S16(5)/PW(14) Q6-S19(3)/PW(23)-S20(4)/PW(24)
	3	Q1-S1(7)/PW(17)-S2(6)/PW(22)-S3(3)/PW(23) Q2-S5(4)/PW(19) Q3-S7(6)/PW(16) Q4-S15(7)/PW(17)
	4	Q1-S1(3)/PW(23)-S2(7)/PW(17) Q2-S4(3)/PW(21)-S5(8)7PW(18) Q3-S8(2)/PW(16)-S9(6)/PW(26) Q6-S20(11)/PW(17)
	5	Q1-S1(4)/PW(21)-S2(6)/PW(17) Q2-S5(8)/PW(18) Q3-S8(5)/PW(24)-S9(6)/PW(26) Q6-S20(8)/PW(27)
80	1	Q1-S1(8)/PW(19)-S2(5)/PW(25)-S3(4)/PW(19) Q2-S6(8)/PW(18)-S7(9)/PW(19) Q4-S10(5)/PW(15)-S12(2)/PW(12) Q5-S15(6)/PW(16)-S16(5)/PW(15) Q6-S20(5)/PW(15)
	2	Q1-S1(7)/PW(23)-S2(4)/PW(17) Q2-S4(5)/PW(19)-S5(7)/PW(18)-S6(6)/PW(16) Q3-S8(5)/PW(16)-S9(7)/PW(7) Q5-S16(4)/PW(24)-S17(3)/PW(25) Q6-S18(5)/PW(25)-S19(6)/PW(23)-S20(4)/PW(14)
	3	Q1-S1(6)/PW(18)-S3(7)/PW(23) Q2-S5(7)/PW(19) Q3-S7(5)/PW(17) Q4-S15(6)/PW(25)-S16(8)/PW(18)
	4	Q1-S1(6)/PW(16)-S2(8)/PW(18)-S3(5)/PW(25) Q2-S4(4)/PW(24)-S5(7)/PW(17)-S6(6)/PW(26) Q3-S8(6)/PW(16)-S9(5)/PW(25) Q6-S19(5)/PW(24)-S20(7)/PW(19)
	5	Q1-S1(7)/PW(15)-S2(6)/PW(20) Q2-S5(7)/PW(21) Q3-S8(4)/PW(18)-S9(6)/PW(16) Q6-S19(8)/PW(19)

Table A2. Part demands and movement costs between cells.

Part	Part Demands					Movement Costs between Cells				
	1. Period	2. Period	3. Period	4. Period	5. Period	1. Period	2. Period	3. Period	4. Period	5. Period
1	150	90	80	70	65	45	40	35	30	40
2	80	75	70	75	70	35	50	45	40	45
3	40	35	30	50	45	30	47	40	37	47
4	75	70	65	85	80	34	48	40	38	48
5	80	75	70	100	95	52	40	55	40	45
6	120	100	90	85	80	55	50	55	50	55
7	60	55	50	65	60	37	45	40	35	45
8	50	45	40	55	50	36	45	35	35	40
9	85	75	70	85	75	33	50	44	40	45
10	90	70	65	90	85	40	35	30	30	35
11	90	75	70	45	40	55	43	40	33	38
12	60	55	50	50	45	32	52	50	36	42
13	50	45	40	40	35	35	45	45	40	45
14	55	50	45	55	50	38	40	50	48	55
15	70	60	55	65	60	45	45	40	39	49
16	110	90	85	70	65	47	42	40	32	45
17	100	80	75	55	50	45	55	50	45	48
18	50	45	40	40	38	40	38	35	32	38
19	65	50	45	55	50	33	45	40	39	42
20	80	75	70	80	75	52	45	40	35	45
21	70	65	60	60	55	40	50	55	50	55
22	100	95	85	90	75	45	55	50	50	55
23	60	55	50	50	45	40	37	35	33	38
24	55	50	45	55	50	43	38	35	30	35
25	70	60	55	65	60	40	30	35	32	38
26	50	45	40	40	35	45	45	40	36	40
27	90	85	80	90	80	55	55	50	48	55
28	45	40	35	70	65	40	45	45	43	53
29	55	50	45	80	75	33	40	40	38	45
30	90	80	75	90	85	40	45	45	40	45
31	120	85	80	60	55	45	38	30	29	45
32	90	85	80	100	95	46	55	50	48	52
33	100	75	70	85	80	32	52	50	45	55
34	85	70	65	80	70	35	48	45	40	45
35	80	70	65	80	70	52	45	35	30	40
36	150	125	115	105	100	41	38	50	45	55
37	90	85	80	80	75	37	52	40	35	45
38	40	35	30	65	60	33	45	45	40	45
39	45	40	35	40	35	43	40	40	35	48
40	90	80	75	60	55	40	43	40	39	42
41	90	65	60	60	50	47	38	35	30	35
42	70	65	60	65	60	45	43	40	35	38
43	60	55	50	75	70	40	45	40	35	40
44	95	60	55	65	60	42	48	45	40	45
45	70	60	50	65	60	51	50	55	50	55
46	150	115	105	75	70	43	35	35	30	39
47	70	65	55	80	70	37	40	40	35	42
48	60	55	45	75	65	33	42	45	42	45
49	75	70	60	80	70	35	46	50	46	50
50	80	70	50	90	80	48	43	45	43	53
51	120	115	95	75	70	43	35	30	30	45
52	70	65	60	60	55	37	53	55	53	55
53	50	45	40	40	35	30	45	45	40	45

Table A2. *Cont.*

Part	Part Demands					Movement Costs between Cells				
	1. Period	2. Period	3. Period	4. Period	5. Period	1. Period	2. Period	3. Period	4. Period	5. Period
54	65	60	55	55	50	34	40	40	35	45
55	70	60	55	55	55	52	45	55	45	50
56	50	45	40	40	37	45	48	40	38	40
57	90	70	60	80	75	34	48	45	44	45
58	70	65	60	75	73	40	45	55	45	48
59	65	60	55	65	64	38	40	35	30	35
60	70	65	60	60	58	52	45	40	38	42
61	130	105	100	95	90	47	36	40	36	40
62	80	65	60	60	55	30	47	45	43	45
63	80	70	60	75	70	35	39	35	33	35
64	70	60	55	70	65	30	41	40	31	35
65	85	70	65	75	70	50	45	45	40	45
66	130	85	80	80	75	40	52	50	42	48
67	80	75	65	70	65	38	55	55	50	55
68	70	65	60	85	80	37	45	40	38	42
69	85	80	70	90	85	40	40	45	40	45
70	90	80	70	75	70	45	50	55	50	55
71	110	95	90	70	65	40	45	50	45	48
72	90	85	80	65	60	35	52	55	52	55
73	70	65	60	60	55	30	45	40	38	42
74	85	50	45	40	35	38	44	45	40	45
75	80	70	60	60	55	55	43	40	38	42
76	60	55	50	65	60	40	50	50	45	55
77	70	65	60	95	90	30	55	55	50	55
78	80	75	70	105	100	40	39	35	36	40
79	65	60	55	70	65	35	49	45	42	45
80	70	55	50	55	45	40	50	50	45	55

Appendix B

Table A3. Optimal routes for the goal programming, ε-constraint, and AUGMECON methods.

Parts	Optimal Route for Goal Programming	Optimal Route for ε-Constraint	Optimal Route for AUGMECON
1	$x_{111}, x_{211}, x_{311}, x_{411}, x_{511}$	$x_{111}, x_{211}, x_{311}, x_{411}, x_{511}$	$x_{111}, x_{211}, x_{311}, x_{411}, x_{511}$
2	$x_{123}, x_{222}, x_{323}, x_{422}, x_{523}$	$x_{122}, x_{222}, x_{323}, x_{422}, x_{523}$	$x_{122}, x_{222}, x_{323}, x_{423}, x_{523}$
3	$x_{135}, x_{235}, x_{335}, x_{435}, x_{535}$	$x_{135}, x_{235}, x_{335}, x_{435}, x_{535}$	$x_{135}, x_{235}, x_{335}, x_{435}, x_{535}$
4	$x_{142}, x_{242}, x_{342}, x_{443}, x_{542}$	$x_{142}, x_{242}, x_{342}, x_{443}, x_{542}$	$x_{142}, x_{242}, x_{342}, x_{443}, x_{542}$
5	$x_{155}, x_{255}, x_{355}, x_{455}, x_{555}$	$x_{155}, x_{255}, x_{355}, x_{455}, x_{555}$	$x_{155}, x_{255}, x_{355}, x_{455}, x_{555}$
6	$x_{165}, x_{265}, x_{365}, x_{465}, x_{565}$	$x_{165}, x_{265}, x_{365}, x_{465}, x_{565}$	$x_{165}, x_{265}, x_{365}, x_{465}, x_{565}$
7	$x_{173}, x_{273}, x_{373}, x_{473}, x_{573}$	$x_{173}, x_{273}, x_{373}, x_{473}, x_{573}$	$x_{173}, x_{273}, x_{373}, x_{473}, x_{573}$
8	$x_{185}, x_{285}, x_{385}, x_{483}, x_{585}$	$x_{185}, x_{285}, x_{385}, x_{483}, x_{585}$	$x_{185}, x_{285}, x_{385}, x_{483}, x_{585}$
9	$x_{192}, x_{295}, x_{395}, x_{492}, x_{592}$	$x_{195}, x_{295}, x_{395}, x_{494}, x_{592}$	$x_{195}, x_{295}, x_{395}, x_{494}, x_{592}$
10	$x_{1,10,4}, x_{2,10,1}, x_{3,10,1}, x_{4,10,1}, x_{5,10,5}$	$x_{1,10,5}, x_{2,10,5}, x_{3,10,1}, x_{4,10,1}, x_{5,10,5}$	$x_{1,10,5}, x_{2,10,5}, x_{3,10,1}, x_{4,10,1}, x_{5,10,5}$
11	$x_{1,11,3}, x_{2,11,3}, x_{3,11,3}, x_{4,11,3}, x_{5,11,3}$	$x_{1,11,3}, x_{2,11,3}, x_{3,11,3}, x_{4,11,3}, x_{5,11,3}$	$x_{1,11,3}, x_{2,11,3}, x_{3,11,3}, x_{4,11,3}, x_{5,11,3}$
12	$x_{1,12,5}, x_{2,12,5}, x_{3,12,5}, x_{4,12,5}, x_{5,12,5}$	$x_{1,12,5}, x_{2,12,5}, x_{3,12,5}, x_{4,12,5}, x_{5,12,5}$	$x_{1,12,5}, x_{2,12,5}, x_{3,12,5}, x_{4,12,5}, x_{5,12,5}$
13	$x_{1,13,5}, x_{2,13,5}, x_{3,13,5}, x_{4,13,5}, x_{5,13,5}$	$x_{1,13,5}, x_{2,13,5}, x_{3,13,5}, x_{4,13,5}, x_{5,13,5}$	$x_{1,13,5}, x_{2,13,5}, x_{3,13,5}, x_{4,13,5}, x_{5,13,5}$
14	$x_{1,14,3}, x_{2,14,3}, x_{3,14,3}, x_{4,14,3}, x_{5,14,3}$	$x_{1,14,3}, x_{2,14,3}, x_{3,14,3}, x_{4,14,3}, x_{5,14,3}$	$x_{1,14,3}, x_{2,14,3}, x_{3,14,3}, x_{4,14,3}, x_{5,14,3}$
15	$x_{1,15,5}, x_{2,15,5}, x_{3,15,2}, x_{4,15,2}, x_{5,15,2}$	$x_{1,15,5}, x_{2,15,5}, x_{3,15,2}, x_{4,15,2}, x_{5,15,2}$	$x_{1,15,5}, x_{2,15,5}, x_{3,15,2}, x_{4,15,2}, x_{5,15,2}$

Table A3. *Cont.*

Parts	Optimal Route for Goal Programming	Optimal Route for ε-Constraint	Optimal Route for AUGMECON
16	$x_{1,16,2}, x_{2,16,1}, x_{3,16,1}, x_{4,16,1}, x_{5,16,3}$	$x_{1,16,2}, x_{2,16,1}, x_{3,16,1}, x_{4,16,1}, x_{5,16,3}$	$x_{1,16,2}, x_{2,16,1}, x_{3,16,1}, x_{4,16,1}, x_{5,16,3}$
17	$x_{1,17,5}, x_{2,17,5}, x_{3,17,5}, x_{4,17,5}, x_{5,17,5}$	$x_{1,17,5}, x_{2,17,5}, x_{3,17,5}, x_{4,17,5}, x_{5,17,5}$	$x_{1,17,5}, x_{2,17,5}, x_{3,17,5}, x_{4,17,5}, x_{5,17,5}$
18	$x_{1,18,4}, x_{2,18,4}, x_{3,18,4}, x_{4,18,4}, x_{5,18,4}$	$x_{1,18,4}, x_{2,18,4}, x_{3,18,4}, x_{4,18,4}, x_{5,18,4}$	$x_{1,18,4}, x_{2,18,4}, x_{3,18,4}, x_{4,18,4}, x_{5,18,4}$
19	$x_{1,19,5}, x_{2,19,5}, x_{3,19,5}, x_{4,19,5}, x_{5,19,5}$	$x_{1,19,5}, x_{2,19,5}, x_{3,19,5}, x_{4,19,5}, x_{5,19,5}$	$x_{1,19,5}, x_{2,19,5}, x_{3,19,5}, x_{4,19,5}, x_{5,19,5}$
20	$x_{1,20,5}, x_{2,20,5}, x_{3,20,5}, x_{4,20,5}, x_{5,20,5}$	$x_{1,20,5}, x_{2,20,5}, x_{3,20,5}, x_{4,20,5}, x_{5,20,5}$	$x_{1,20,5}, x_{2,20,5}, x_{3,20,5}, x_{4,20,5}, x_{5,20,5}$
21	$x_{1,21,5}, x_{2,21,5}, x_{3,21,5}, x_{4,21,5}, x_{5,21,5}$	$x_{1,21,5}, x_{2,21,5}, x_{3,21,5}, x_{4,21,5}, x_{5,21,5}$	$x_{1,21,5}, x_{2,21,5}, x_{3,21,5}, x_{4,21,5}, x_{5,21,5}$
22	$x_{1,22,4}, x_{2,22,4}, x_{3,22,4}, x_{4,22,4}, x_{5,22,4}$	$x_{1,22,4}, x_{2,22,4}, x_{3,22,4}, x_{4,22,4}, x_{5,22,4}$	$x_{1,22,4}, x_{2,22,4}, x_{3,22,4}, x_{4,22,4}, x_{5,22,4}$
23	$x_{1,23,5}, x_{2,23,5}, x_{3,23,5}, x_{4,23,5}, x_{5,23,5}$	$x_{1,23,5}, x_{2,23,5}, x_{3,23,5}, x_{4,23,5}, x_{5,23,5}$	$x_{1,23,5}, x_{2,23,5}, x_{3,23,5}, x_{4,23,5}, x_{5,23,5}$
24	$x_{1,24,5}, x_{2,24,5}, x_{3,24,5}, x_{4,24,5}, x_{5,24,5}$	$x_{1,24,5}, x_{2,24,5}, x_{3,24,5}, x_{4,24,5}, x_{5,24,5}$	$x_{1,24,5}, x_{2,24,5}, x_{3,24,5}, x_{4,24,5}, x_{5,24,5}$
25	$x_{1,25,3}, x_{2,25,3}, x_{3,25,3}, x_{4,25,3}, x_{5,25,3}$	$x_{1,25,3}, x_{2,25,3}, x_{3,25,3}, x_{4,25,3}, x_{5,25,3}$	$x_{1,25,3}, x_{2,25,3}, x_{3,25,3}, x_{4,25,3}, x_{5,25,3}$
26	$x_{1,26,4}, x_{2,26,4}, x_{3,26,4}, x_{4,26,4}, x_{5,26,4}$	$x_{1,26,4}, x_{2,26,4}, x_{3,26,4}, x_{4,26,4}, x_{5,26,4}$	$x_{1,26,4}, x_{2,26,4}, x_{3,26,4}, x_{4,26,4}, x_{5,26,4}$
27	$x_{1,27,4}, x_{2,27,4}, x_{3,27,4}, x_{4,27,4}, x_{5,27,4}$	$x_{1,27,4}, x_{2,27,4}, x_{3,27,4}, x_{4,27,4}, x_{5,27,4}$	$x_{1,27,4}, x_{2,27,4}, x_{3,27,4}, x_{4,27,4}, x_{5,27,4}$
28	$x_{1,28,3}, x_{2,28,3}, x_{3,28,3}, x_{4,28,3}, x_{5,28,3}$	$x_{1,28,3}, x_{2,28,3}, x_{3,28,3}, x_{4,28,3}, x_{5,28,3}$	$x_{1,28,3}, x_{2,28,3}, x_{3,28,3}, x_{4,28,3}, x_{5,28,3}$
29	$x_{1,29,1}, x_{2,29,1}, x_{3,29,1}, x_{4,29,1}, x_{5,29,1}$	$x_{1,29,1}, x_{2,29,1}, x_{3,29,1}, x_{4,29,1}, x_{5,29,1}$	$x_{1,29,1}, x_{2,29,1}, x_{3,29,1}, x_{4,29,1}, x_{5,29,1}$
30	$x_{1,30,4}, x_{2,30,4}, x_{3,30,4}, x_{4,30,4}, x_{5,30,4}$	$x_{1,30,4}, x_{2,30,4}, x_{3,30,4}, x_{4,30,4}, x_{5,30,4}$	$x_{1,30,4}, x_{2,30,4}, x_{3,30,4}, x_{4,30,4}, x_{5,30,4}$
31	$x_{1,31,5}, x_{2,31,3}, x_{3,31,3}, x_{4,31,3}, x_{5,31,5}$	$x_{1,31,5}, x_{2,31,3}, x_{3,31,3}, x_{4,31,3}, x_{5,31,5}$	$x_{1,31,5}, x_{2,31,3}, x_{3,31,3}, x_{4,31,3}, x_{5,31,5}$
32	$x_{1,32,5}, x_{2,32,5}, x_{3,32,5}, x_{4,32,5}, x_{5,32,5}$	$x_{1,32,5}, x_{2,32,5}, x_{3,32,5}, x_{4,32,5}, x_{5,32,5}$	$x_{1,32,5}, x_{2,32,5}, x_{3,32,5}, x_{4,32,5}, x_{5,32,5}$
33	$x_{1,33,2}, x_{2,33,4}, x_{3,33,4}, x_{4,33,4}, x_{5,33,2}$	$x_{1,33,2}, x_{2,33,4}, x_{3,33,4}, x_{4,33,4}, x_{5,33,2}$	$x_{1,33,2}, x_{2,33,4}, x_{3,33,4}, x_{4,33,4}, x_{5,33,2}$
34	$x_{1,34,4}, x_{2,34,1}, x_{3,34,1}, x_{4,34,1}, x_{5,34,4}$	$x_{1,34,4}, x_{2,34,1}, x_{3,34,1}, x_{4,34,1}, x_{5,34,4}$	$x_{1,34,4}, x_{2,34,1}, x_{3,34,1}, x_{4,34,1}, x_{5,34,4}$
35	$x_{1,35,3}, x_{2,35,3}, x_{3,35,3}, x_{4,35,3}, x_{5,35,3}$	$x_{1,35,3}, x_{2,35,3}, x_{3,35,3}, x_{4,35,3}, x_{5,35,3}$	$x_{1,35,3}, x_{2,35,3}, x_{3,35,3}, x_{4,35,3}, x_{5,35,3}$
36	$x_{1,36,1}, x_{2,36,1}, x_{3,36,1}, x_{4,36,1}, x_{5,36,1}$	$x_{1,36,1}, x_{2,36,1}, x_{3,36,1}, x_{4,36,1}, x_{5,36,1}$	$x_{1,36,1}, x_{2,36,1}, x_{3,36,1}, x_{4,36,1}, x_{5,36,1}$
37	$x_{1,37,2}, x_{2,37,4}, x_{3,37,3}, x_{4,37,4}, x_{5,37,3}$	$x_{1,37,2}, x_{2,37,2}, x_{3,37,3}, x_{4,37,2}, x_{5,37,3}$	$x_{1,37,2}, x_{2,37,2}, x_{3,37,3}, x_{4,37,3}, x_{5,37,3}$
38	$x_{1,38,3}, x_{2,38,3}, x_{3,38,3}, x_{4,38,3}, x_{5,38,3}$	$x_{1,38,3}, x_{2,38,3}, x_{3,38,3}, x_{4,38,3}, x_{5,38,3}$	$x_{1,38,3}, x_{2,38,3}, x_{3,38,3}, x_{4,38,3}, x_{5,38,3}$
39	$x_{1,39,3}, x_{2,39,3}, x_{3,39,3}, x_{4,39,3}, x_{5,39,3}$	$x_{1,39,3}, x_{2,39,3}, x_{3,39,3}, x_{4,39,3}, x_{5,39,3}$	$x_{1,39,3}, x_{2,39,3}, x_{3,39,3}, x_{4,39,3}, x_{5,39,3}$
40	$x_{1,40,5}, x_{2,40,5}, x_{3,40,5}, x_{4,40,1}, x_{5,40,5}$	$x_{1,40,5}, x_{2,40,5}, x_{3,40,5}, x_{4,40,5}, x_{5,40,5}$	$x_{1,40,5}, x_{2,40,5}, x_{3,40,5}, x_{4,40,5}, x_{5,40,5}$
41	$x_{1,41,1}, x_{2,41,1}, x_{3,41,1}, x_{4,41,1}, x_{5,41,1}$	$x_{1,41,1}, x_{2,41,1}, x_{3,41,1}, x_{4,41,1}, x_{5,41,1}$	$x_{1,41,1}, x_{2,41,1}, x_{3,41,1}, x_{4,41,1}, x_{5,41,1}$
42	$x_{1,42,3}, x_{2,42,4}, x_{3,42,3}, x_{4,42,4}, x_{5,42,3}$	$x_{1,42,4}, x_{2,42,4}, x_{3,42,3}, x_{4,42,4}, x_{5,42,3}$	$x_{1,42,4}, x_{2,42,4}, x_{3,42,3}, x_{4,42,3}, x_{5,42,3}$
43	$x_{1,43,3}, x_{2,43,3}, x_{3,43,3}, x_{4,43,3}, x_{5,43,3}$	$x_{1,43,3}, x_{2,43,3}, x_{3,43,3}, x_{4,43,3}, x_{5,43,3}$	$x_{1,43,3}, x_{2,43,3}, x_{3,43,3}, x_{4,43,3}, x_{5,43,3}$
44	$x_{1,44,3}, x_{2,44,3}, x_{3,44,3}, x_{4,44,3}, x_{5,44,3}$	$x_{1,44,3}, x_{2,44,3}, x_{3,44,3}, x_{4,44,3}, x_{5,44,3}$	$x_{1,44,3}, x_{2,44,3}, x_{3,44,3}, x_{4,44,3}, x_{5,44,3}$
45	$x_{1,45,5}, x_{2,45,5}, x_{3,45,5}, x_{4,45,1}, x_{5,45,5}$	$x_{1,45,5}, x_{2,45,5}, x_{3,45,5}, x_{4,45,5}, x_{5,45,5}$	$x_{1,45,5}, x_{2,45,5}, x_{3,45,5}, x_{4,45,5}, x_{5,45,5}$
46	$x_{1,46,5}, x_{2,46,5}, x_{3,46,5}, x_{4,46,5}, x_{5,46,5}$	$x_{1,46,5}, x_{2,46,3}, x_{3,46,5}, x_{4,46,5}, x_{5,46,5}$	$x_{1,46,5}, x_{2,46,3}, x_{3,46,5}, x_{4,46,5}, x_{5,46,5}$
47	$x_{1,47,2}, x_{2,47,2}, x_{3,47,2}, x_{4,47,2}, x_{5,47,2}$	$x_{1,47,2}, x_{2,47,2}, x_{3,47,2}, x_{4,47,2}, x_{5,47,2}$	$x_{1,47,2}, x_{2,47,2}, x_{3,47,2}, x_{4,47,2}, x_{5,47,2}$
48	$x_{1,48,3}, x_{2,48,3}, x_{3,48,3}, x_{4,48,3}, x_{5,48,3}$	$x_{1,48,3}, x_{2,48,3}, x_{3,48,3}, x_{4,48,3}, x_{5,48,3}$	$x_{1,48,3}, x_{2,48,3}, x_{3,48,3}, x_{4,48,3}, x_{5,48,3}$
49	$x_{1,49,2}, x_{2,49,2}, x_{3,49,2}, x_{4,49,2}, x_{5,49,2}$	$x_{1,49,2}, x_{2,49,2}, x_{3,49,2}, x_{4,49,2}, x_{5,49,2}$	$x_{1,49,2}, x_{2,49,2}, x_{3,49,2}, x_{4,49,2}, x_{5,49,2}$
50	$x_{1,50,4}, x_{2,50,4}, x_{3,50,4}, x_{4,50,3}, x_{5,50,4}$	$x_{1,50,4}, x_{2,50,5}, x_{3,50,4}, x_{4,50,3}, x_{5,50,4}$	$x_{1,50,4}, x_{2,50,5}, x_{3,50,4}, x_{4,50,3}, x_{5,50,4}$
51	$x_{1,51,3}, x_{2,51,3}, x_{3,51,3}, x_{4,51,3}, x_{5,51,3}$	$x_{1,51,3}, x_{2,51,3}, x_{3,51,3}, x_{4,51,3}, x_{5,51,3}$	$x_{1,51,3}, x_{2,51,3}, x_{3,51,3}, x_{4,51,3}, x_{5,51,3}$
52	$x_{1,52,5}, x_{2,52,5}, x_{3,52,5}, x_{4,52,5}, x_{5,52,5}$	$x_{1,52,5}, x_{2,52,5}, x_{3,52,5}, x_{4,52,5}, x_{5,52,5}$	$x_{1,52,5}, x_{2,52,5}, x_{3,52,5}, x_{4,52,5}, x_{5,52,5}$
53	$x_{1,53,5}, x_{2,53,5}, x_{3,53,5}, x_{4,53,5}, x_{5,53,5}$	$x_{1,53,5}, x_{2,53,5}, x_{3,53,5}, x_{4,53,5}, x_{5,53,5}$	$x_{1,53,5}, x_{2,53,5}, x_{3,53,5}, x_{4,53,5}, x_{5,53,5}$
54	$x_{1,54,5}, x_{2,54,3}, x_{3,54,3}, x_{4,54,3}, x_{5,54,3}$	$x_{1,54,5}, x_{2,54,3}, x_{3,54,3}, x_{4,54,3}, x_{5,54,3}$	$x_{1,54,5}, x_{2,54,3}, x_{3,54,3}, x_{4,54,3}, x_{5,54,3}$
55	$x_{1,55,2}, x_{2,55,2}, x_{3,55,2}, x_{4,55,2}, x_{5,55,2}$	$x_{1,55,2}, x_{2,55,2}, x_{3,55,2}, x_{4,55,2}, x_{5,55,2}$	$x_{1,55,2}, x_{2,55,2}, x_{3,55,2}, x_{4,55,2}, x_{5,55,2}$
56	$x_{1,56,2}, x_{2,56,1}, x_{3,56,1}, x_{4,56,1}, x_{5,56,2}$	$x_{1,56,2}, x_{2,56,1}, x_{3,56,1}, x_{4,56,1}, x_{5,56,3}$	$x_{1,56,2}, x_{2,56,1}, x_{3,56,1}, x_{4,56,1}, x_{5,56,3}$
57	$x_{1,57,5}, x_{2,57,5}, x_{3,57,5}, x_{4,57,5}, x_{5,57,5}$	$x_{1,57,5}, x_{2,57,5}, x_{3,57,5}, x_{4,57,5}, x_{5,57,5}$	$x_{1,57,5}, x_{2,57,5}, x_{3,57,5}, x_{4,57,5}, x_{5,57,5}$
58	$x_{1,58,4}, x_{2,58,4}, x_{3,58,4}, x_{4,58,4}, x_{5,58,4}$	$x_{1,58,4}, x_{2,58,4}, x_{3,58,4}, x_{4,58,4}, x_{5,58,4}$	$x_{1,58,4}, x_{2,58,4}, x_{3,58,4}, x_{4,58,4}, x_{5,58,4}$
59	$x_{1,59,5}, x_{2,59,5}, x_{3,59,5}, x_{4,59,5}, x_{5,59,5}$	$x_{1,59,5}, x_{2,59,5}, x_{3,59,5}, x_{4,59,5}, x_{5,59,5}$	$x_{1,59,5}, x_{2,59,5}, x_{3,59,5}, x_{4,59,5}, x_{5,59,5}$
60	$x_{1,60,5}, x_{2,60,5}, x_{3,60,5}, x_{4,60,5}, x_{5,60,5}$	$x_{1,60,5}, x_{2,60,5}, x_{3,60,5}, x_{4,60,5}, x_{5,60,5}$	$x_{1,60,5}, x_{2,60,5}, x_{3,60,5}, x_{4,60,5}, x_{5,60,5}$
61	$x_{1,61,2}, x_{2,61,2}, x_{3,61,5}, x_{4,61,5}, x_{5,61,5}$	$x_{1,61,5}, x_{2,61,2}, x_{3,61,5}, x_{4,61,5}, x_{5,61,5}$	$x_{1,61,5}, x_{2,61,2}, x_{3,61,5}, x_{4,61,5}, x_{5,61,5}$
62	$x_{1,62,4}, x_{2,62,4}, x_{3,62,4}, x_{4,62,4}, x_{5,62,4}$	$x_{1,62,4}, x_{2,62,4}, x_{3,62,4}, x_{4,62,4}, x_{5,62,4}$	$x_{1,62,4}, x_{2,62,4}, x_{3,62,4}, x_{4,62,4}, x_{5,62,4}$
63	$x_{1,63,5}, x_{2,63,5}, x_{3,63,5}, x_{4,63,5}, x_{5,63,5}$	$x_{1,63,5}, x_{2,63,5}, x_{3,63,5}, x_{4,63,5}, x_{5,63,5}$	$x_{1,63,5}, x_{2,63,5}, x_{3,63,5}, x_{4,63,5}, x_{5,63,5}$

Table A3. *Cont.*

Parts	Optimal Route for Goal Programming	Optimal Route for ε-Constraint	Optimal Route for AUGMECON
64	$x_{1,64,5}, x_{2,64,5}, x_{3,64,5}, x_{4,64,5}, x_{5,64,5}$	$x_{1,64,5}, x_{2,64,5}, x_{3,64,5}, x_{4,64,5}, x_{5,64,5}$	$x_{1,64,5}, x_{2,64,5}, x_{3,64,5}, x_{4,64,5}, x_{5,64,5}$
65	$x_{1,65,3}, x_{2,65,3}, x_{3,65,3}, x_{4,65,3}, x_{5,65,3}$	$x_{1,65,3}, x_{2,65,3}, x_{3,65,3}, x_{4,65,3}, x_{5,65,3}$	$x_{1,65,3}, x_{2,65,3}, x_{3,65,3}, x_{4,65,3}, x_{5,65,3}$
66	$x_{1,66,4}, x_{2,66,4}, x_{3,66,4}, x_{4,66,4}, x_{5,66,4}$	$x_{1,66,4}, x_{2,66,4}, x_{3,66,4}, x_{4,66,4}, x_{5,66,4}$	$x_{1,66,4}, x_{2,66,4}, x_{3,66,4}, x_{4,66,4}, x_{5,66,4}$
67	$x_{1,67,4}, x_{2,67,4}, x_{3,67,4}, x_{4,67,4}, x_{5,67,4}$	$x_{1,67,4}, x_{2,67,4}, x_{3,67,4}, x_{4,67,4}, x_{5,67,4}$	$x_{1,67,4}, x_{2,67,4}, x_{3,67,4}, x_{4,67,4}, x_{5,67,4}$
68	$x_{1,68,3}, x_{2,68,3}, x_{3,68,3}, x_{4,68,3}, x_{5,68,3}$	$x_{1,68,3}, x_{2,68,3}, x_{3,68,3}, x_{4,68,3}, x_{5,68,3}$	$x_{1,68,3}, x_{2,68,3}, x_{3,68,3}, x_{4,68,3}, x_{5,68,3}$
69	$x_{1,69,3}, x_{2,69,1}, x_{3,69,3}, x_{4,69,1}, x_{5,69,3}$	$x_{1,69,3}, x_{2,69,1}, x_{3,69,3}, x_{4,69,1}, x_{5,69,3}$	$x_{1,69,3}, x_{2,69,1}, x_{3,69,3}, x_{4,69,1}, x_{5,69,3}$
70	$x_{1,70,4}, x_{2,70,4}, x_{3,70,4}, x_{4,70,4}, x_{5,70,4}$	$x_{1,70,4}, x_{2,70,4}, x_{3,70,4}, x_{4,70,4}, x_{5,70,4}$	$x_{1,70,4}, x_{2,70,4}, x_{3,70,4}, x_{4,70,4}, x_{5,70,4}$
71	$x_{1,71,3}, x_{2,71,3}, x_{3,71,3}, x_{4,71,3}, x_{5,71,3}$	$x_{1,71,3}, x_{2,71,3}, x_{3,71,3}, x_{4,71,3}, x_{5,71,3}$	$x_{1,71,3}, x_{2,71,3}, x_{3,71,3}, x_{4,71,3}, x_{5,71,3}$
72	$x_{1,72,5}, x_{2,72,5}, x_{3,72,5}, x_{4,72,5}, x_{5,72,5}$	$x_{1,72,5}, x_{2,72,5}, x_{3,72,5}, x_{4,72,5}, x_{5,72,5}$	$x_{1,72,5}, x_{2,72,5}, x_{3,72,5}, x_{4,72,5}, x_{5,72,5}$
73	$x_{1,73,2}, x_{2,73,4}, x_{3,73,4}, x_{4,73,4}, x_{5,73,2}$	$x_{1,73,2}, x_{2,73,4}, x_{3,73,4}, x_{4,73,4}, x_{5,73,2}$	$x_{1,73,2}, x_{2,73,4}, x_{3,73,4}, x_{4,73,4}, x_{5,73,2}$
74	$x_{1,74,4}, x_{2,74,1}, x_{3,74,1}, x_{4,74,1}, x_{5,74,4}$	$x_{1,74,4}, x_{2,74,1}, x_{3,74,1}, x_{4,74,1}, x_{5,74,4}$	$x_{1,74,4}, x_{2,74,1}, x_{3,74,1}, x_{4,74,1}, x_{5,74,4}$
75	$x_{1,75,3}, x_{2,75,3}, x_{3,75,3}, x_{4,75,3}, x_{5,75,3}$	$x_{1,75,3}, x_{2,75,3}, x_{3,75,3}, x_{4,75,3}, x_{5,75,3}$	$x_{1,75,3}, x_{2,75,3}, x_{3,75,3}, x_{4,75,3}, x_{5,75,3}$
76	$x_{1,76,1}, x_{2,76,1}, x_{3,76,1}, x_{4,76,1}, x_{5,76,1}$	$x_{1,76,2}, x_{2,76,1}, x_{3,76,1}, x_{4,76,1}, x_{5,76,1}$	$x_{1,76,2}, x_{2,76,1}, x_{3,76,1}, x_{4,76,1}, x_{5,76,1}$
77	$x_{1,77,3}, x_{2,77,2}, x_{3,77,3}, x_{4,77,2}, x_{5,77,3}$	$x_{1,77,2}, x_{2,77,2}, x_{3,77,3}, x_{4,77,2}, x_{5,77,3}$	$x_{1,77,2}, x_{2,77,2}, x_{3,77,3}, x_{4,77,3}, x_{5,77,3}$
78	$x_{1,78,3}, x_{2,78,3}, x_{3,78,3}, x_{4,78,3}, x_{5,78,3}$	$x_{1,78,3}, x_{2,78,3}, x_{3,78,3}, x_{4,78,3}, x_{5,78,3}$	$x_{1,78,3}, x_{2,78,3}, x_{3,78,3}, x_{4,78,3}, x_{5,78,3}$
79	$x_{1,79,3}, x_{2,79,3}, x_{3,79,3}, x_{4,79,3}, x_{5,79,3}$	$x_{1,79,3}, x_{2,79,3}, x_{3,79,3}, x_{4,79,3}, x_{5,79,3}$	$x_{1,79,3}, x_{2,79,3}, x_{3,79,3}, x_{4,79,3}, x_{5,79,3}$
80	$x_{1,80,5}, x_{2,80,5}, x_{3,80,5}, x_{4,80,5}, x_{5,80,5}$	$x_{1,80,5}, x_{2,80,5}, x_{3,80,5}, x_{4,80,5}, x_{5,80,5}$	$x_{1,80,5}, x_{2,80,5}, x_{3,80,5}, x_{4,80,5}, x_{5,80,5}$

References

1. Coca, G.; Castrillón, O.D.; Ruiz, S.; Mateo-Sanz, J.M.; Jiménez, L. Sustainable evaluation of environmental and occupational risks scheduling flexible job shop manufacturing systems. *J. Clean. Prod.* **2019**, *209*, 146–168. [CrossRef]
2. Singh, S.; Olugu, E.U.; Musa, S.N. Development of sustainable manufacturing performance evaluation expert system for small and medium enterprises. *Procedia CIRP* **2016**, *40*, 608–613. [CrossRef]
3. Ahmad, S.; Wong, K.Y.; Rajoo, S. Sustainability indicators for manufacturing sectors A literature survey and maturity analysis from the triple-bottom line perspective. *J. Manuf. Technol. Manag.* **2019**, *30*, 312–334. [CrossRef]
4. Sutherland, J.W.; Richter, J.S.; Hutchins, M.J.; Dornfeld, D.; Dzombak, R.; Mangold, J.; Robinson, S.; Hauschild, M.Z.; Bonou, A.; Schönsleben, P.; et al. The role of manufacturing in affecting the social dimension of sustainability. *CIRP Ann. Manuf. Technol.* **2016**, *65*, 689–712. [CrossRef]
5. Galal, N.M.; Moneim, A.F.A. A mathematical programming approach to the optimal sustainable product mix for the process industry. *Sustainability* **2015**, *7*, 13085–13103. [CrossRef]
6. Rajak, S.; Vinodh, S. Application of fuzzy logic for social sustainability performance evaluation: A case study of an Indian automotive component manufacturing organization. *J. Clean. Prod.* **2015**, *108*, 1184–1192. [CrossRef]
7. Vimal, K.E.K.; Vinodh, S.; Raja, A. Modelling, assessment and deployment of strategies for ensuring sustainable shielded metal arc welding process-a case study. *J. Clean. Prod.* **2015**, *93*, 364–377. [CrossRef]
8. Lin, C.J.; Belis, T.T.; Kuo, T.C. Ergonomics-based factors or criteria for the evaluation of sustainable product manufacturing. *Sustainability* **2019**, *11*, 4955. [CrossRef]
9. Mengistu, A.T.; Panizzolo, R. Tailoring sustainability indicators to small and medium enterprises for measuring industrial sustainability performance. *Meas. Bus. Excell.* **2023**, *27*, 54–70. [CrossRef]
10. Giret, A.; Trentesaux, D.; Prabhu, V. Sustainability in manufacturing operations scheduling: A state of the art review. *J. Manuf. Syst.* **2015**, *37*, 126–140. [CrossRef]
11. Niakan, F.; Baboli, A.; Moyaux, T.; Botta-Genoulaz, V. A bi-objective model in sustainable dynamic cell formation problem with skill-based worker assignment. *J. Manuf. Syst.* **2016**, *38*, 46–62. [CrossRef]
12. Jain, A.K.; Kasilingam, R.G.; Bhole, S.D. Joint consideration of cell formation and tool provisioning problems in flexible manufacturing systems. *Comput. Ind. Eng.* **1991**, *20*, 271–277. [CrossRef]
13. Alhourani, F. Cellular manufacturing system design considering machines reliability and parts alternative process routings. *Int. J. Prod. Res.* **2016**, *54*, 846–863. [CrossRef]
14. Ghodsi, R.; Mostafayi, S.; Mansouri, Z.; Bakhtiari, M. Designing a bi-objective integrating mathematical model for dynamic sustainable cellular manufacturing systems considering production planning. *J. Appl. Mech. Eng.* **2015**, *4*, 184.
15. Aljuneidi, T.; Bulgak, A.A. A mathematical model for designing reconfigurable cellular hybrid manufacturing-remanufacturing systems. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 1585–1596. [CrossRef]
16. Mehdizadeh, E.; Niaki, S.V.D.; Rahimi, V. A vibration damping optimization algorithm for solving a new multi-objective dynamic cell formation problem with workers training. *Comput. Ind. Eng.* **2016**, *101*, 35–52. [CrossRef]

17. Niakan, F.; Baboli, A.; Moyaux, T.; Botta-Genoulaz, V. A new multi-objective mathematical model for dynamic cell formation under demand and cost uncertainty considering social criteria. *Appl. Math. Model.* **2016**, *40*, 2674–2691. [CrossRef]
18. Imran, M.; Kang, C.; Lee, Y.H.; Jahanzaib, M.; Aziz, H. Cell formation in a cellular manufacturing system using simulation integrated hybrid genetic algorithm. *Comput. Ind. Eng.* **2017**, *105*, 123–135. [CrossRef]
19. Arghish, O.; Tavakkoli-Moghaddam, R.; Shahandeh-Nookabadi, A.; Rezaeian, J. Comparison among four calibrated meta-heuristic algorithms for solving a type-2 fuzzy cell formation problem considering economic and environmental criteria. *Adv. Prod. Eng. Manag.* **2018**, *13*, 5–17. [CrossRef]
20. Iqbal, A.; Al-Ghamdi, K.A. Energy-efficient cellular manufacturing system: Eco-friendly revamping of machine shop configuration. *Energy* **2018**, *163*, 863–872. [CrossRef]
21. Kumar, R.; Singh, S.P. Simulated annealing-based embedded meta-heuristic approach to solve bi-objective robust stochastic sustainable cellular layout. *Glob. J. Flex. Syst. Manag.* **2018**, *19*, 69–93. [CrossRef]
22. Raoofpanah, H.; Ghezavati, V.; Tavakkoli-Moghaddam, R. Solving a new robust green cellular manufacturing problem with environmental issues under uncertainty using Benders decomposition. *Eng. Optim.* **2019**, *51*, 1229–1250. [CrossRef]
23. Telegraphi, A.H.; Bulgak, A.A. A mathematical model for designing sustainable cellular remanufacturing systems. *J. Ind. Syst. Eng.* **2021**, *13*, 98–123.
24. Forghani, K.; Fatemi Ghomi, S.M.T.; Kia, R. Group layout design of manufacturing cells incorporating assembly and energy aspects. *Eng. Optim.* **2022**, *54*, 770–785. [CrossRef]
25. Jafarzadeh, J.; Khalili, H.A.; Shoja, N. A multiobjective optimization model for a dynamic and sustainable cellular manufacturing system under uncertainty. *Comput. Intell. Neurosci.* **2022**, *2022*, 1334081. [CrossRef] [PubMed]
26. Aryanezhad, M.B.; Deljoo, V.; Mirzapour Al-e-hashem, S.M.J. Dynamic cell formation and the worker assignment problem: A new model. *Int. J. Adv. Manuf. Technol.* **2009**, *41*, 329–342. [CrossRef]
27. Fan, J.; Feng, D. Design of cellular manufacturing system with quasi-dynamic dual resource using multi-objective GA. *Int. J. Prod. Res.* **2013**, *51*, 4134–4154. [CrossRef]
28. Bagheri, M.; Bashiri, M. A new mathematical model towards the integration of cell formation with operator assignment and inter-cell layout problems in a dynamic environment. *Appl. Math. Model.* **2014**, *38*, 1237–1254. [CrossRef]
29. Azadeh, A.; Pashapour, S.; Zadeh, S.A. Designing a cellular manufacturing system considering decision style, skill and job security by NSGA-II and response surface methodology. *Int. J. Prod. Res.* **2016**, *54*, 6825–6847. [CrossRef]
30. Mehdizadeh, E.; Rahimi, V. An integrated mathematical model for solving dynamic cell formation problem considering operator assignment and inter/intra cell layouts. *Appl. Soft Comput.* **2016**, *42*, 325–341. [CrossRef]
31. Nouri, H. Development of a comprehensive model and BFO algorithm for a dynamic cellular manufacturing system. *Appl. Math. Model.* **2016**, *40*, 1514–1531. [CrossRef]
32. Zohrevand, A.M.; Rafiei, H.; Zohrevand, A.H. Multi-objective dynamic cell formation problem: A stochastic programming approach. *Comput. Ind. Eng.* **2016**, *98*, 323–332. [CrossRef]
33. Sadeghi, S.; Forghani, M.A.; Seidi, M. Integrated dynamic cell formation with operator assignment and inter-cell layout problems: A mathematical model. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2017**, *231*, 1658–1669. [CrossRef]
34. Zhang, J.; Zhou, B. Design an incremental cell formation problem considering human factor. In Proceedings of the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 3–5 October 2017; pp. 559–563.
35. Delgoshaei, A.; Mirzazadeh, A.; Ali, A. A hybrid ant colony system and tabu search algorithm for the production planning of dynamic cellular manufacturing systems while confronting uncertain costs. *Braz. J. Oper. Prod. Manag.* **2018**, *15*, 499–516. [CrossRef]
36. Fahmy, S.A. MILP model and a genetic algorithm for the integrated cell formation, layout and worker assignment problem. In Proceedings of the 2019 IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, 15–17 February 2019; pp. 117–121.
37. Sakhaii, M.; Tavakkoli-Moghaddam, R.; Bagheri, M.; Vatani, B. A robust optimization approach for an integrated dynamic cellular manufacturing system and production planning with unreliable machines. *Appl. Math. Model.* **2016**, *40*, 169–191. [CrossRef]
38. Feng, H.; Da, W.; Xi, L.; Pan, E.; Xia, T. Solving the integrated cell formation and worker assignment problem using particle swarm optimization and linear programming. *Comput. Ind. Eng.* **2017**, *110*, 126–137. [CrossRef]
39. Shafiee-Gol, S.; Kia, R.; Kazemi, M.; Tavakkoli-Moghaddam, R.; Darmian, S.M. A mathematical model to design dynamic cellular manufacturing systems in multiple plants with production planning and location–allocation decisions. *Soft Comput.* **2021**, *25*, 3931–3954. [CrossRef]
40. Felfel, H.; Ayadi, O.; Masmoudi, F. A decision-making approach for a multi-objective multisite supply network planning problem. *Int. J. Comput. Integr. Manuf.* **2016**, *29*, 754–767. [CrossRef]
41. Mavrotas, G. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Appl. Math. Comput.* **2009**, *213*, 455–465. [CrossRef]

42. Mahdavi, I.; Teymourian, E.; Baher, N.T.; Kayvanfar, V. An integrated model for solving cell formation and cell layout problem simultaneously considering new situations. *J. Manuf. Syst.* **2013**, *32*, 655–663. [CrossRef]
43. Yadollahi, M.S.; Mahdavi, I.; Paydar, M.M.; Jouzdani, J. Design a bi-objective mathematical model for cellular manufacturing systems considering variable failure rate of machines. *Int. J. Prod. Res.* **2014**, *52*, 7401–7415. [CrossRef]

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Applied Sciences Editorial Office
E-mail: applsci@mdpi.com
www.mdpi.com/journal/applsci

MDPI

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-4408-1